# Towards an extensible privacy analysis framework for Smart Homes

A. Girish♣♦, J. Tapiador♦, S. Matic♠ and N. Vallina-Rodriguez♣♥
♣ IMDEA Networks Institute, ♦ Universidad Carlos III de Madrid, ♠ IMDEA Software Institute, ♥ AppCensus

## ABSTRACT

The IoT ecosystem is an intricate and complex network of stakeholders that includes platforms, developers, ad networks and cloud providers. However, the ability of smart home platforms and devices to interact and exchange data, together with the data-driven business models adopted by most IoT stakeholders open the ground for unknown and unexpected privacy risks. Existing black-box testing approaches to audit IoT platforms cannot identify data dissemination through side- and covert-channels, and for this reason they are not well suited for rich execution environments where a wide range of devices and applications can co-operate using multiple network protocols and interfaces. This poster proposes ImposTer, a cost-effective and extensible privacy framework for exhaustively testing the IoT ecosystem. Our framework is able to capture, model and emulate horizontal interactions that occur across the different devices in a consumer household.

**ACM Reference Format:**
A. Girish♣♦, J. Tapiador♦, S. Matic♠ and N. Vallina-Rodriguez♣♥. 2022. Towards an extensible privacy analysis framework for Smart Homes. In *Proceedings of the 22nd ACM Internet Measurement Conference (IMC '22), October 25–27, 2022, Nice, France.* ACM, New York, NY, USA, 3 pages. https://doi.org/10.1145/3517745.3563023

## 1 INTRODUCTION

The research community has shown that the proliferation of smart devices with rich sensors such as GPS and microphone along with persistent Internet connectivity has created a new type and persistent set of privacy threats [2, 7]. Unfortunately, existing analysis techniques are not able to automatically test privacy behaviors in modern smart home devices. Prior research efforts tend to assume that IoT devices are monolithic, deterministic, and self-contained entities that can be tested in isolation, or using a limited number of devices in expensive IoT labs [1, 7]. However, this approach does not correspond to reality: IoT devices and applications deployed on home networks interact with other devices and applications, and integrate third-party components as well as non-deterministic AI features. Therefore, privacy-intrusive behaviors may only manifest when a device interacts with a specific set of apps and peer devices in the local network.

Therefore, a key question still remains unanswered: *what sort of behaviors do smart devices exhibit in any arbitrary rich execution environments with a multitude of components and non-deterministic inputs? More importantly, what potential privacy and security violations do we miss?* To answer to these questions, we propose ImposTer, a novel cost-effective and extensible privacy analysis framework for testing privacy properties of smart home products, automatically and at scale. ImposTer is currently an early-stage working prototype, and it contains multiple elements that facilitate the discovery of potential privacy-intrusive behaviors across different platforms. Our framework combines static and dynamic analysis techniques with novel traffic interception and network fuzzing methods.

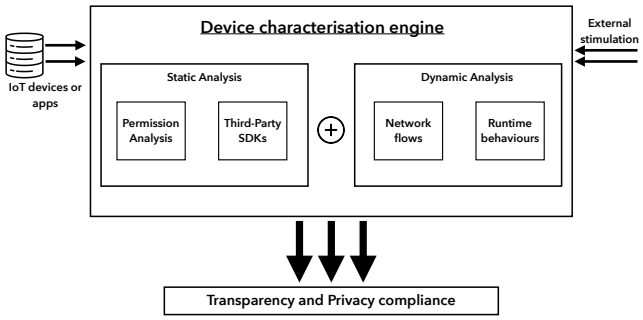


**Figure 1: ImposTer architecture, including different system components and the hybrid analysis approach for auditing privacy compliance of IoT devices.**

## 2 SYSTEM OVERVIEW

Figure 1 shows ImposTer's architecture design. We combine and extend well-established static and dynamic analysis techniques to analyze app behavior. This combination allows capturing and reasoning about IoT device behaviors with a higher certainty, coverage and visibility than when the two techniques are used separately.

**Static Analysis.** We perform static analysis on Android-based IoT apps (both companion apps and apps that run on the host device, including Android-based devices like Smart TVs and other products), to gather signals that can be retrieved without having to execute the app. We extract essential characteristics such as manifest information, runtime permissions, unique strings (including URLs) and code-signatures which are immune to obfuscation methods. For Android and Android TV apps, we also perform flow and taint analysis. Additionally, we use LibRadar [3] to identify third-party libraries embedded in apps, and to attribute behaviors to either first or third-party components.

**Dynamic analysis.** ImposTer also dynamically analyses the behavior of IoT devices introduced to the environment. To facilitate runtime monitoring, we rely on Frida [6], a well-known dynamic code instrumentation toolkit for dynamic analysis. We develop custom Frida scripts, that monitor runtime and network behaviors of


*IMC '22, October 25–27, 2022, Nice, France*

ACM ISBN 978-1-4503-9259-4/22/10.
https://doi.org/10.1145/3517745.3563023

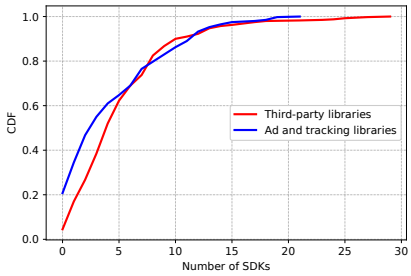


**Figure 2: Distribution of SDKs across smart TV apps.**

Android based IoT apps, including networking activity and access to privacy-sensitive APIs. To test apps at scale we use a dedicated smart User Interface (UI) exerciser, developed in-house. Our UI exerciser simulates real user interactions by generating UI test inputs based on a state transition model generated on-the-fly. In the future we plan to integrate techniques for automatically testing smart speakers by issuing voice commands, and simulate physical interactions on smart appliances.

**Network monitoring.** In addition to our dynamic analysis instrumentation, we perform network monitoring from two complementary vantage points: (*i*) a Wi-Fi access point with a transparent proxy equipped with mitmproxy [4]; and (*ii*) a honeypot that supports popular IoT protocols (e.g., UPnP, mDNS, HTTP(s)), APIs (*e.g.,* Spotify Connect, Google's Nearby) and wireless interfaces (e.g., Bluetooth, ZigBee). Currently, the honeypot is connected to the access point and it passively captures traffic from services, device discoveries, scans, and requests. As future work, we plan to leverage the honeypot traffic to develop an automated network-level fuzzer that can emulate different smart home devices and services. The fuzzer will allow ImposTer to stimulate behaviors and detect privacy violations that happen through side- and covert-channels.

## 3 PRELIMINARY RESULTS

We empirically measure and validate the properties and capabilities of our working prototype. We run ImposTer in a testbed with real IoT devices such as smart speakers (*e.g.,* Amazon Echo, Google Home), IoT gateways to control smart appliances (*e.g.,* Samsung smartThings, Starling Home Hub) and smart home surveillance (*e.g.,* Ring ecosystem). We inspect with static and dynamic analysis a corpus of 415 TV apps and 16 companion apps of the IoT devices in our dataset collected from Google's official Play Store. All logs (*i.e.,* network and runtime) are collected and passively analyzed, and augmented with the outcome of our static analysis pipeline.

**Third-party services.** Modern software developers tend to rely on third-party services (or SDKs) to speed up their development process but also for integrating advertising and tracking services (ATS). Figure 2 illustrates that half of the apps that we analyzed have from two, up to seven different ATS SDKs. Similarly to the mobile ecosystem, smart TV apps often integrate analytics services such as Firebase and Admob owned by Google. In addition, some apps contain ad libraries such as Tapjoy and Vungle that specialize in connecting smart TV services.

**Inter device scanning.** Dynamic analysis offers a complementary perspective. Mobile apps make use of multicast UPnP SSDP (239.255.255.250:1900) and Netbios scans to legitimately discover peripheral-controlling devices. Moreover, we found our smartThings hub sending 1-byte UDP packets on random ports to receive as reply ICMP packets from co-located devices that had such ports open; while Amazon devices (e.g., Fire TV and Alexa) communicate over UDP:55444, possibly for spatial perception [8] or synchronizing time. The ability to discover and upload to the cloud network and device fingerprints can have adverse privacy effects as it can be used to geo-locate users trough their WiFi AP, but also to infer the socio-economic level of the household and social structures.

**PII dissemination.** ImposTer detects transmission and leaks of identifiers such as Android ID, Device ID *etc.* over the network. We found that 62 (15.5%) apps share non-resettable Device IDs over the Internet. Whereas 7 apps combine the resettable AAID with the persistent Android ID, a practice prohibited by Google Terms of Service as it defeats users' privacy choices.

## 4 RELATED WORK

A large-scale study of IoT data exposure is often hard because accessing a large number of IoT devices is difficult, challenging to inspect and expensive to get. For example, existing black-box testing approaches designed by Moghaddam *et al.* [5] and Ren *et al.* [7] focus on a small set of IoT devices deployed in lab environments. Both works utilize network traffic generated by semi-automated interaction with the device to identify the types of data transferred over the Internet and the parties that receive the data. Our work shares many of the design and techniques proposed in prior work, however, ImposTer automatically uncovers unwanted data exposure by *smartly* simulating real user interactions and proposes a new systematic approach to discover cross-device communications.

## 5 CONCLUSIONS

This poster presents and showcases a working prototype of ImposTer, a cost-effective framework for automatic and unified privacy testing of IoT platforms. ImposTer stands apart from prior work by considering the interconnected nature of IoT devices, and for its ability to monitor and simulate smart home devices and platform behaviors at the network level, independently from device type or the vendor.

## 6 ACKNOWLEDGEMENTS

This project has been funded by the EU's H2020 Program (TRUST aWARE Project, Grant Agreement No. 101021377) and the Spanish Ministry of Science (ODIO Project, PID2019-111429RB-C22); The opinions, findings, and conclusions or recommendations expressed are those of the authors and do not necessarily reflect those of any of the funders.

## REFERENCES

[1] J. S. Edu, X. Ferrer-Aran, J. M. Such, and G. Suarez-Tangi. 2021. SkillVet: Automated Traceability Analysis of Amazon Alexa Skills. (2021). http://arxiv.org/abs/2103.02637
[2] FTC. 2017. VIZIO case. https://www.ftc.gov/news-events/press-releases/2017/02/vizio-pay-22-million-ftc-state-new-jersey-settle-charges-it.
[3] Z. Ma, H. Wang, Y. Guo, and X. Chen. 2016. LibRadar: fast and accurate detection of third-party libraries in Android apps. In *Proceedings of International Conference on Software Engineering (ICSE).*
[4] Mitmproxy Project. 2010. mitmproxy. https://mitmproxy.org.

[5] H. M. Moghaddam, G. Acar, B. Burgess, A. Mathur, D. Y. Huang, N. Feamster, E. W. Felten, P. Mittal, and A. Narayanan. 2019. Watching You Watch: The Tracking Ecosystem of Over-the-Top TV Streaming Devices. In *Proceedings of Conference on Computer and Communications Security (CCS)*.
[6] O. A. V. Ravnås. 2021. Frida.re. https://frida.re.
[7] J. Ren, D. J. Dubois, D. Choffnes, A. M. Mandalari, R. Kolcun, and H. Haddadi. 2019. Information Exposure From Consumer IoT Devices: A Multidimensional, Network-Informed Measurement Approach. In *Proceedings of Internet Measurement Conference (IMC)*.
[8] T. Warren. 2018. Amazon is making it easier for all Alexa devices to work together. https://www.theverge.com/2018/7/25/17613832/amazon-alexa-echo-spatial-perception.