The Ultrasound Tracking Ecosystem

V. Mavroudis[1], S. Hao[2], Y. Fratantonio[2], F. Maggi[3], C. Kruegel[2] & G. Vigna[2]

The ultrasound tracking ecosystem is a relatively new and emerging set of technologies that use inaudible beacons to track users and devices. Our work is the first comprehensive security analysis of this little known ecosystem.

Introduction

In the last two years, the marketing industry started to show a fast increasing interest in technologies for user cross-device tracking, proximity tracking, and their derivative monetization schemes. To meet these demands, ultrasound-tracking, a new technology based on ultrasounds, has emerged and is already utilized in a number of different real-world applications. This new tracking technology comes with numerous desirable features (e.g., easy to deploy, inaudible to humans), but alarmingly until now no comprehensive analysis of its security had been conducted. In this report, we summarize the results of our security analysis on the ultrasound tracking ecosystem, demonstrate the practical security and privacy risks that arise with its adoption, and introduce a number of countermeasures.

uBeacons

Ultrasound beacons (uBeacons, in short) are high-frequency audio tags that can be emitted and captured by most commercial speakers and microphones, and are not audible by humans. These beacons are at the core of all ultrasound tracking products currently at the market and encode a small sequence of characters and symbols. In most cases, this sequence serves as an identifier used to fetch content from an external server or to pair two devices together, as we discuss next.

From a technical perspective, an ultrasound beacon has a duration of only few seconds, usually around 5. However, the exact method of encoding the information, varies greatly depending on the requirements of the application (e.g., range requirements, information volume). In the great majority of cases, the spectrum between 18000Hz and 20000Hz is divided in smaller chunks, and each one corresponds to a symbol or character. In our experiments, uBeacons had very low error rate in distances up to ∼7 meters, however, they cannot penetrate through physical obstacles (e.g., walls, doors).

Currently, there is no standard or specification for uBeacons, and hence each company designs its own beacon encoding formats and protocols. As a result, there are multiple incompatible frameworks, providing varying levels of security.

Cross Device Tracking

The main advantage of the ultrasound technology compared to already existing solutions is that it does not require any specialized equipment (unlike wifi and bluetooth), while it remains inaudible to humans. For this reason, the technology is already utilized for cross-device tracking, that is currently the "holy grail" of marketers as it allows them to track the user's activities across different devices to provide relevant, more targeted services. For example, knowing that John Doe who has just watched a TV ad is the same John Doe that is now browsing the Internet from his smartphone to find a birthday gift allows advertisers to display relevant ads that can result in higher conversion into purchases.

XDT techniques are currently employed by major advertisement networks and offering varying degrees of device linking precision. For instance, the most accurate technique requires the user to sign in with his/her account at the advertiser's service from all the devices that he/she owns. This model is suitable for social media providers (e.g., facebook), where the users are incentivized to login to the system. However, in most cases this is not a reasonable expectation, and hence the advertisers are using alternative XDT techniques.

One such technique is ultrasound Cross-device Tracking (uXDT), where ultrasound beacons (uBeacons) are embedded into websites or TV ads and get picked up by advertisement SDKs embedded in smartphone apps. Its main advantage is that it offers very high accuracy, without requiring from the user to purposefully link his/her devices (e.g., login to a single service from all of them). However, it requires an uXDT framework to be installed on the user's mobile device. Such frameworks usually come incorporated in advertising 'software development kits' (SDKs) that

[1] v.mavroudis@cs.ucl.ac.uk

[2] {shuanghao, yanick, chris, vigna}@cs.ucsb.edu

[3] federico.maggi@polimi.it

app developers add to their products to increase their revenue.

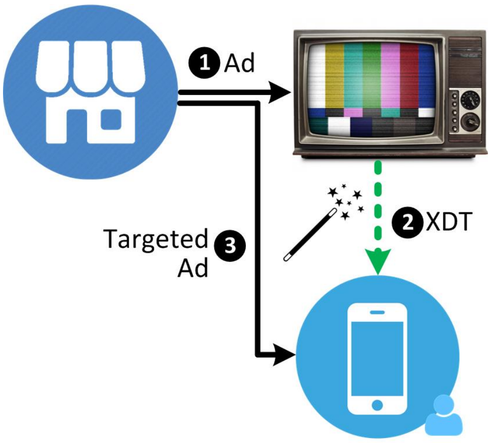

Figure 1: Illustration of how advertisers use ultrasound cross-device tracking to target users across multiple devices.

Figure 1 demonstrates how advertisers use uXDT to target users across multiple devices. At first, the advertiser pushes an ultrasound-enabled ad to the user (❶). Such an ad can be a TV ad, or an ad on a website that the user visits using his/her laptop. Once the ad is displayed, a short sequence of high-frequency (i.e., ultrasonic) tones is emitted from the speakers of the device and is promptly captured by the uXDT framework on the user's smartphone (❷). To achieve this, the uXDT framework runs on the background and periodically access the device's microphone to listen for ultrasound sequences. Once such a sequence is captured, the framework extracts the unique identifier of the ad from it and reports it back to the advertiser, along with some unique device/user identifiers. The advertiser then uses this information to infer the user's interests, build a user profile, and finally push targeted ads to the user's device (❸).

Other Use Cases

Apart from uXDT, there are also other tracking technologies based on ultrasounds: One of these is *device pairing*, where a device *A* uses ultrasounds to broadcast a random PIN to nearby devices, and trigger a pairing once a device B submits the correct PIN back (usually through the Internet). One such example is Google Cast, a popular app developed by Google, that reportedly utilizes ultrasound beacons to facilitate device pairing between mobile devices and the Google Chromecast.

Moreover, ultrasounds are also utilized in proximity marketing, where they serve as a mean to provide location-specific content and track in-store user behaviour. For this purpose, stores and other venues setup ultrasound emitters (regular speakers usually) in multiple locations. Once a user with an ultrasound-enabled app visits the store, the app picks these beacons up and reports them to the company's backend. Subsequently, the company can use this information to study the user's in-store behaviour, provide real-time notifications for products in proximity, and offer reward points for visiting the store.

In addition to the above use cases, ultrasounds have also been used for audience measurements and analytics. For instance, they offer an effective way to measure the number of viewers of a specific TV ad, and their reactions (e.g., switching channels).

Privacy & Security Considerations

The ultrasound tracking ecosystem remained almost unknown to the general public until recently, when Silverpush faced the nemesis of the security community and the regulators (e.g., the Federal Trade Commission) for its controversial tracking techniques. However, this was treated as an isolated security incident and little became known about other companies with ultrasound-enabled products and the ecosystem as a whole. To cover this gap, we analysed frameworks from various companies and identified two serious security problems that affect the ecosystem as a whole and are not specific to the individual frameworks.

To begin with, the security model of ultrasound tracking technologies relies on the transmitting range of ultrasounds, assuming no physical proximity of an attacker. However, this is very flimsy: not only ultrasounds can travel reliably for a few meters, but there are various ways in which an attacker can get "virtually close" to the beacon receiver. For instance, an attacker can use a small code snippet (e.g., embedded in a web page or in an ad campaign that the attacker controls) that, when loaded, will automatically reproduce one or more attacker-chosen audio beacons. We call such a snippet a *beacon trap*.

Using this technique an attacker can easily mount *beacon-injection* or *beacon-replay* attacks, which will cause the nearby ultrasound-enabled devices to capture and report beacons of her choice. For instance, for the beacon-injection attack, imagine an attacker equipped with a simple beacon-emitting device (e.g., smartphone) walking into Starbucks at peak hour. As a result, all customers with an ultrasound-enabled app installed on their devices will be receiving the beacons and unknowingly forward them to the advertiser's backend. In the "replay" variation of this attack, the adversary captures and replays *existing* beacons (e.g., to influence an analytics campaign).

Using these three attack primitives (traps, injection, replay) an adversary can craft more complex attacks. For instance, a Tor-deanonymization attack can be launched by setting up a beacon trap in a (malicious) Tor hidden service. When visited, the trap will emit the uBeacons and, if the victim is using a uXDT (or other tracking) framework on a non-Tor-connected device, the user will be de-anonymized thanks to the "pairing" created by the tracking provider. In this regard, a state-sponsored attacker or the tracking provider itself can mount this attack. Another example is the profile-corruption attack, where the attacker uses the beacon-injection technique to pollute the ad profile of the users. In a similar vein, an attacker could cause an information leak of the victim's interests. However, in the last attack the extend of the leak depends on the profiling techniques used by each company.

The second security problem identified is the violation of the principle of least privilege. Any app that wants to employ ultrasound-based mechanisms needs to gain full access to the device's microphone. This clearly violates the least privilege principle, as the app has now access to all audible frequencies. Currently, there is no way to gain access only to the ultrasound spectrum, as existing versions of the Android framework do not expose any mechanism that allows fine-grained control over the device's microphone. This has several negative repercussions: A malicious developer could claim access to the microphone for ultrasound-pairing purposes, and then use it to spy on the user (e.g., to record the audio). On the other hand, any ultrasound-enabled app risks to be perceived as "potentially malicious" by the users. To make things worse, we observed wide discrepancies between the practices followed by companies when it comes to informing the users and providing opt-out options. For instance, in some cases no notice or opt-out option is given to the user (apart from the mandatory microphone permission request), while in other cases the framework developers simply advice the developers to provide opt-out options, but do not enforce it.

Moreover, to better understand the extend of the problem we studied the popularity of ultrasound-enabled products found in the market. Our findings indicate that the ecosystem is growing fast with new companies and products appearing at a fast pace. Fortunately, the number of affected users seems to be still relatively low (at most few millions), and this highlights the need for immediate action before the user base further expands.

Currently, there are ∼10 companies offering tracking solutions based on ultrasounds, with the great majority of them providing frameworks for proximity marketing. Proximity marketing based on ultrasounds gained a tremendous traction in the last few years, mainly because it makes use of the existing audio infrastructure and requires no additional equipment (unlike Bluetooth). On the other hand, the only company providing an uXDT framework was forced to withdraw it from the US market after the backslash from the security community. We believe that this (along with the sophisticated infrastructure required for uXDT) disincentivized other companies from providing similar services as the users are very suspicious towards uXDT and any security mischiefs would permanently tarnish the reputation of the company.

All in all, our security examination revealed critical and exploitable security problems in the ecosystem. Fortunately, the concept of ultrasound-tracking is relatively new, and only a limited number of users could be affected by malicious attacks (for now). However, it is paramount to take immediate action to address these security shortcomings (without reducing the usability of the technology), as these technologies gain traction fast and the user-base expands.

Where do We Go From Here?

To address the aforementioned issues, we developed a set of countermeasures aiming to provide protection to the users in the short and medium-term.

The first one is an extension for the Google Chrome browser, which filters out all ultrasounds from the audio output of the websites loaded by the user. The extension actively prevents web pages from emitting inaudible sounds, and thus completely thwarts any unsolicited attempts of ultrasound-tracking. To achieve this, it mediates all audio outputs of the page and filters out the frequencies which fall within the range used by inaudible beacons. Its only limitation is that, due to some shortcomings of HTML5, it cannot directly filter a few legacy/obsolete technologies, such as Flash player. Furthermore, we have also developed a patch for the Android permission system that allows finer-grained control over the audio channel, and forces applications to declare their intention to capture sound from the inaudible spectrum. This will properly separate the permissions for listening to audible sound and sound in the high-frequency (i.e., ultrasound) spectrum, and will enable the end users to selectively filter the ultrasound frequencies out of the signal acquired by the smartphone microphone.

While we believe that the above countermeasures significantly increase the security and awareness of end users, we argue that the ultrasounds ecosystem can be made secure only with the standardization of the ultrasound beacon format. Once this process is completed, APIs for handling uBeacons can be implemented in all major operating systems. Such an API should expose calls for uBeacon discovery, processing, generation and emission, similarly to the Bluetooth Low Energy APIs. Thereafter, all ultrasound-enabled apps will need access only to this API, and not to the device's microphone. Thus, solving the problem of over-privilaging that exposed the user's sensitive data to third-party apps. Additionally, app developers that use ultrasound-tracking technologies will no longer risk being deemed as "potentially malicious" by the users. However, to properly enforce the use of this API for all the apps, the ultrasound spectrum must be accessible only to privileged components of the system. To achieve this, the system module handling the microphone should filter out ultrasonic frequencies by default, and only the user should be able to grant access to the spectrum on a per-app basis. This will force all third-party developers to make use of the central API when implementing ultrasound-based functionality.

We believe that the ultrasound-tracking ecosystem has a lot of potential in providing useful services to both users and companies. However, its current realization leaves a lot to be desired in terms of privacy and security, thus exposing the users to unnecessary risks. Our work provides an early warning on these risks, and lays the foundations for the secure use of this set of technologies.