Bluetooth Beacons and How to Make Users Untraceable

Yanyan Ren
Brown University

Yuchen Yang
Brown University

1 Background

Bluetooth Beacons are devices that continuously send Bluetooth Low Energy(BLE) signals that cover the range of about 80 meters [4]. The signals can be received by nearby mobile devices and then processed by the mobile applications. The accuracy of localization is about 5 meters. Note that the GPS has a similar accuracy of localization, but it doesn't work well indoors [7]. After detecting the signal, the phone sends the UUID of the beacon to a server. The server responds according to the ID of the beacon and the phone gets the notifications. Since the beacon only needs to broadcasts a small amount of information (mostly UUID), the device can be quite small and use little energy. It is often used by companies to track customers inside stores. See figure 1 for more details of how beacon, phone and server interact.

There are mainly 2 kinds of protocols used for bluetooth beacons.

iBeacon [6] is developed by Apple and is the most popular protocol used in the area. It does better job for interactivity between iOS devices and iBeacon hardware. For beacons using iBeacon, they have better integrity at identifying beacons and applications can direct the information of beacons to iBeacon servers. Universally unique identifier(UUID) is the only information needed to find the developer server.

Eddystone [2] is developed by Google and is a more flexible one that contains security features and APIs but requires more knowledge from developers [3]. The broadcasted packets not only contain the UUID, but also sensor telemetry(temperature, battery status) and a URL address of the developer's server. This indicates that it does not require specific applications from the developer and can get the response directly from a common beacon application.

2 Plans

To get a general idea of how popular beacons are and where they are used, we first want to walk around Providence with a beacon tracker app such as [1]. The app would pick up any beacon signal the phone picked up, and display the beacon's location and UUID. We plan to visit 3 different kinds of neighborhoods ? Providence Place, Brown campus, and the residential area around Hope street, and see how they differ in beacon usage.

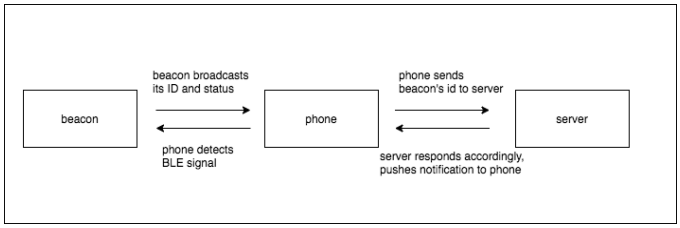

Figure 1: How a beacon works

The second part of our plan is to make a spoofing utility so that a user can be untraceable. Here is one common scenario of beacon usage: in a grocery store, a beacon hidden behind the yogurt counter sends a signal to your phone, then your phone communicates with the server, sending both the beacon information saying that you are looking at yogurts and your shopping history, soon after, a coupon of your favorite yogurt brand is pushed to your phone's grocery app. With our hack, we still want to keep the general functionality of the beacon and the app, but doesn't want to expose the user's personal information. Therefore, in the example mentioned above, you would still get a coupon, but the server wouldn't get any more information than someone is in front of this yogurt counter.

To make this spoofing utility, we plan to put a blackbox between the phone and the server's communication 2. With our blackbox in the middle, the phone gets mapped to a temporary id before talking to the server. Thus, the server can't see any personal information about the phone, but can still responds according to the beacon's information attached. When a packet is sent back from the server, our blackbox matches the temporary id to the phone, and delivers the packet.

To build the blackbox, first we need to find out how to capture a packet sent from the phone. Then we can look into

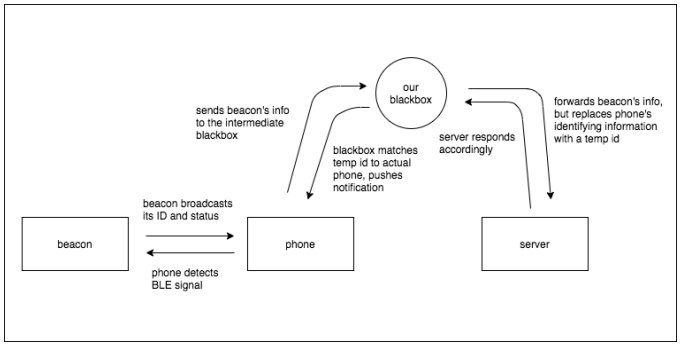

Figure 2: Spoofing with our blackbox

the structure of the packet, and figure out how the beacon info and phone's identifying info are written inside the packets. We need to build a storing structure to match the pair (phone info, temp id). As a stretch goal, we want to make our blackbox secure and trustworthy. Potential solutions include encrypting the phone info to prevent any plaintext being stored, and using secure multi-party computation to avoid placing trust in one single machine.

3 Techniques

To collect the packet information, we can use Tcpdump [10] directly in both Android and iOS systems.

After analyzing some initial packets, Netsed [5] is a handy tool to modify the texts and headers.

For storage, we can look into encrypted key-value storing systems such as Khaled [8] and Trousseau [9].

There could possibly be some other techniques applied during the process of development.

References

[1] Beacon tracker - apps on google play. https://play.google.com/store/apps/details?id=teksun.beacon.tracker&hl=en_US.

[2] Beacons | google developers. https://developers.google.com/beacons.

[3] ibeacon vs eddystone: Which one works better for your pilot project? https://blog.beaconstac.com/2016/01/ibeacon-vs-eddystone/.

[4] Beaconstac. What is a bluetooth beacon? how do ble beacons work? https://www.beaconstac.com/what-is-a-bluetooth-beacon#working.

[5] Canonical. http://manpages.ubuntu.com/manpages/trusty/man1/netsed.1.html.

[6] Apple Inc. ibeacon. https://developer.apple.com/ibeacon/.

[7] Michael Kwet. In stores, secret bluetooth surveillance tracks your every move, Jun 2019. https://www.nytimes.com/interactive/2019/06/14/opinion/bluetooth-wireless-tracking-privacy.html.

[8] Nasser. khaled, Jul 2017. https://github.com/nasser/khaled.

[9] Oleiade. trousseau, Aug 2019. https://github.com/oleiade/trousseau.

[10] Tcpdump. Tcpdump libpcap public repository, Sep 2010. https://www.tcpdump.org/.