Enhancing the Trust Level in Geocaching

Degree programme: Computer Science | Specialisation: IT-Security
Thesis advisor: Dr. Reto Koenig
Expert: Stefan Berner

The last few years have shown an increasing growth in the popularity of Geocaching. However, a few aspects of this evolution are starting to clash with the original system. Many geocache owners are having trouble keeping track of the increasing number of log entries; cheaters are taking advantage of the situation by registering finds online, but not actually finding the geocache on the spot. This project offers a solution that combines logbook and website entries in one log.

What is Geocaching?

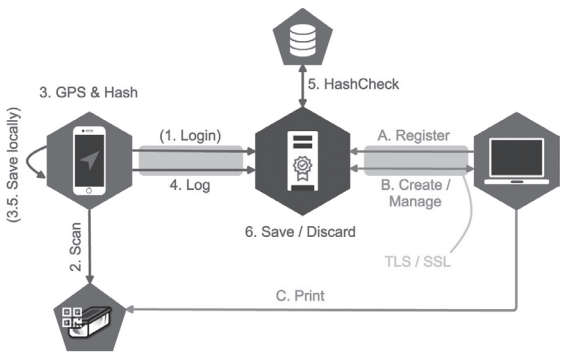
Geocaching is a recreational activity where players hide objects (typically boxes) for other players to find using GPS services. In simple words, players engage in a treasure hunt to find an object which contains a logbook. The logbook serves as a mean of proving a person has found a geocache, as the players need to sign the logbook with username and date upon retrieval.

The problem...

The popularity growth of the last few years has made the concept of comparing logbook to website entries tedious and unpopular. This trend enables cheaters to report a find without having actually found a geocache. The goal of this project is to analyze and find solutions which automate the «logbook to website» entry check for owners. The solution needs to: respect cost boundaries of geocache creation, be applicable to the existing system owned by geocaching.com, and be available on iOS, Android & Windows Mobile.

... and the solution

Most geocachers go on the hunt with their mobile device, as it provides GPS functionalities and (when available) Internet connection. The concept of using the mobile device to collect a piece of information as proof from the geocache is fairly simple, and it's possible to use different hardware to do so (camera, NFC sensor, microphone, ...). Barcodes represent a cheap, fast and easy solution for storing a value inside a geocache. Said value can be read off the barcode and sent to the server as proof of find by almost any type of smartphone. The barcode should store an identifying number (as in the current system) and a verification code that can only be obtained there.



GeoSec System Flowchart

Other noteworthy requirements

- Offline log saving (mobile app)
- Position check, user must be within 1 km of the geocache at scan time
- Verification code revocation
- Log saved entries (scanned **before** revocation) **after** revocation
- Manual logging (in case of geocache damage) by the owner for affected geocachers
- Geocache management portal

Proof of possession

After having scanned and obtained the verification code, it wouldn't be very secure to just send said code in cleartext through the Internet to the centralized server – even over a TLS channel. Custom encryption requires keys, which makes the system more complicated than needed by demanding a key distribution system.
Hashing the verification code together with other public values needed by the server hides sensitive information from unintended ears. The server can compute a second hash with the public values obtained from the client and the verification code stored in its database. If the received hash matches the computed one, the log request is considered valid.

Frameworks & technologies

- Apache Cordova: convert a website into mobile platform apps
- AngularJS: Javascript data-binding framework
- Slim Framework: RESTful backend
- MongoDB: noSQL Database backend
- Framework7: dynamic mobile UI based on the device's OS
- Bootstrap: website frontend UI

Valentina Claudia Lanz
valentina.lanz@me.com