AndroidTracker

Christoph Leitner, Alexandra Jäger

January 26, 2015

Quality and Security Program

University of Innsbruck

QSP TIROL

Task

- Write an Android application that tracks a user
- Runs in background
- Sends GPS location information to Web application
- Uses Wifi to determine location of the device
- Allow remote locking of the device
- Build a Web application that plots the user's activity

App - Introduction

- built for Android 4.3
- use Google Play Services to read GPS data from mobile phone
- use **HTTPS**-POST to send it to server
- information sent:
  - IMEI – unique device identification
  - LAT – GPS coordinates: latitude
  - LONG – GPS coordinates: longitude
  - ACCURACY – GPS coordinates accuracy in meters

App - Detailed Description

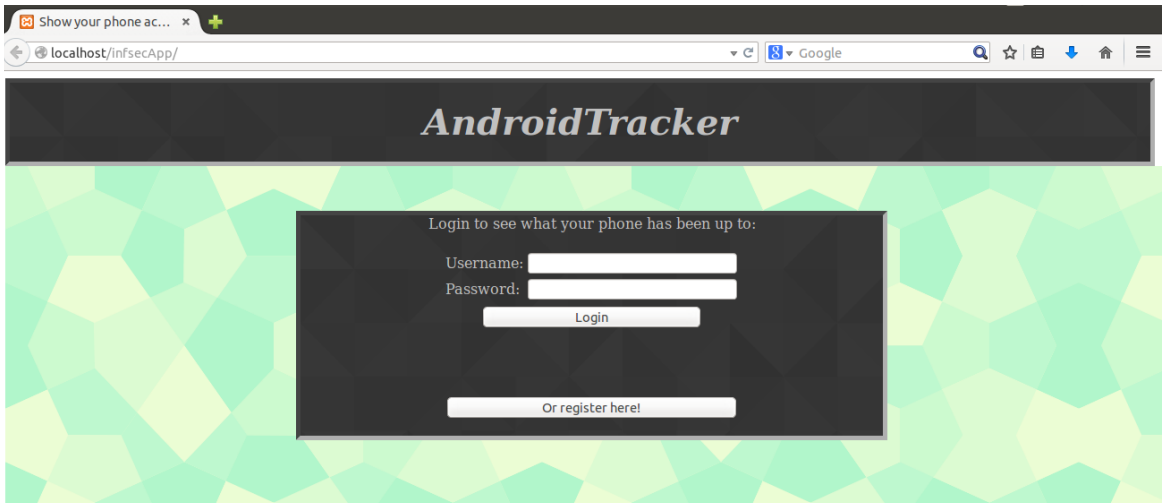
- user can change interval of GPS coordinate transmissions (default is 60 seconds)
- user can enable device administration for the app
  - needed for remote locking and wiping of the device
- for SSL: create own *TrustManager*, that trusts only the certificate of our server

App - Remote Locking & Wipe

- server sends lock command and new password in HTTP response to the POSTed location data
  - JSON encoded
  - `{ "200": { "cmd": "lock", "data": "password"}}`
- therefore aggregation of data points is performed on the server
- for wiping, only the wipe command is sent in HTTP response

Web Application

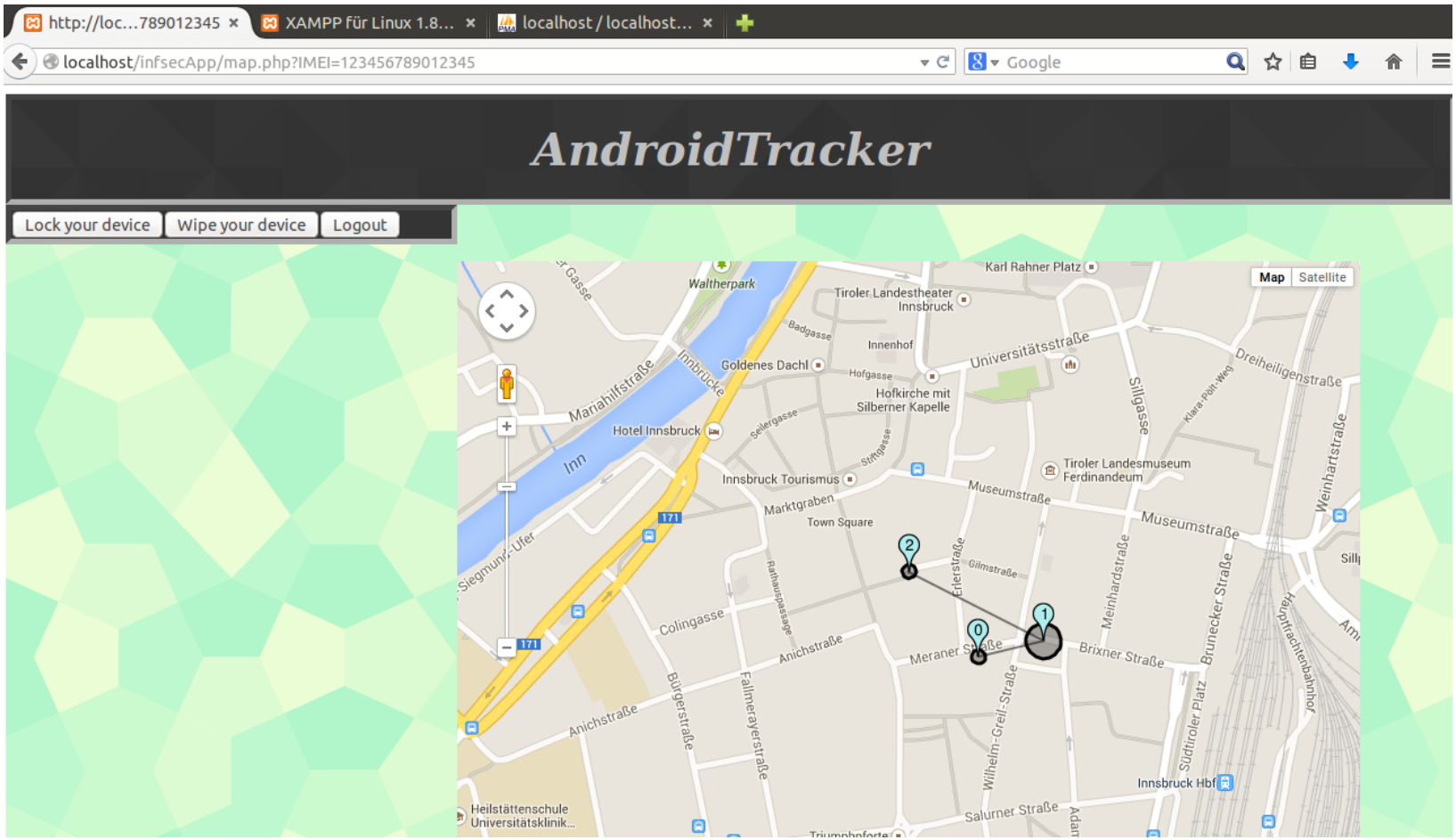
- use XAMPP to set up a local server
- to plot your activity register with a username, a password and your IMEI
- passwords are saved as hash using a salt value



Web Application - Map Features

- locations are displayed using Google Map Markers
- numbers are written in the markers to indicate the order in which the location data has been received
- lines between markers indicate paths
- circle around markers indicate accuracy
- upon mouse over the time-stamp of the location update as well as the accuracy is shown
- map centers around last known location

Web Application - Screenshot



Web Application - Details

- locking and wiping requests are saved in DB and sent to the user with the next arrival of coordinates
- data aggregation:
  - received coordinates are compared to last received ones
  - if difference (in meters) is smaller than a certain threshold (currently 3 meters) the location is not saved in the database, but timestamp of last location is updated

Thank you for your attention!

QUESTIONS?

- `https://github.com/christophL/AndroidTracker`