EURECOM – FALL 2019

WISEC

EURECOM

Sophia Antipolis

GPS SPOOFING

Students

Francesco Galati

Angelo Russi

Teachers

Aurélien Francillon

Giovanni Camurati

INTRODUCTION

Mobile navigation services are being used by billions of users nowadays, while paper maps are definitively deprecated. There is a huge difference between their usage though: while the second ones give to the user a global vision of the area he is travelling through, requiring a continuous attention to what surrounds him (i.e. landmarks, street names, road signs, etc.), the first ones guide the users in a strict passive way. In fact, most navigation systems display a "first-person" view, forcing the focus only on the current road and the next turn: the natural user response consists in losing the attention on the surrounding environment, relying more likely on the voice prompt and the visual instructions (i.e., road shapes, arrows, etc.). But are these navigation systems really reliable and secure? To answer this question we have been studying "All Your GPS Are Belong To Us: Towards Stealthy Manipulation of Road Navigation Systems".

GPS

First, it's necessary to underline an important fact: map applications (i.e., Google Maps, Waze, etc.) depend on GPS inputs, which are way to be reliable. GPS is indeed vulnerable to spoofing attacks, where adversaries can inject falsified GPS signals to control the victim's device. This happens because the channel the GPS satellites are transmitting their signals through (L1, 1575.42 MHz) does not adopt any defence mechanism. GPS spoofing attacks consist of two keys steps:

- the takeover step, that forces the victim GPS receiver to migrate from the legitimate signal to a fake one;
- the manipulation step, during which the GPS receiver is at the mercy of the attacker.

The takeover phase can be brute-forced or smooth. In the former case, the victim loses track of the satellites receiving fake signals at high power. In contrast, the smooth takeover consists of gradually overpowering the original signal after synchronizing with them, causing the desired migration.

THE HACK

We now know that a possible attacker could send fake GPS signals to an unsuspecting user: if an attacker identifies an attacking route that mimics the shape of the route displayed on the map, it becomes possible to trigger navigation instructions that are consistent with the physical environment. Since the driver puts most of his focus to the next instruction and hardly distrusts the navigator, it will be sufficient for the hacker to make sure there is compatibility between the ghost route (displayed on the receiver) and the victim route (actually followed by the driver). When the attack begins, the

malicious signals are sent starting from a first ghost location, that generates a small location drift, forcing the system to recalculate the route to the destination. These behaviours, unless the jump is too big, generally do not cause any suspicion to the user, because it is kind of common to experience GPS malfunction caused by losing GPS signals or wrong positioning. In order to understand the hack better and verify its feasibility we have forced GPS locations in an Android virtual machine to test Google Maps behaviour, probably the most famous navigation app online.

OUR DEMO

Our work has been divided in two different stages, following a simple idea: we could have started using GPS-SDR-SIM to send the fake locations. This tool allows to generate binary files representing the spoofing signals, that we would have subsequently read using GNSS-SDR, a software defined receiver. Then, using the GPX file just created by the receiver, we would have virtually set the read locations into an Android virtual machine run using QEMU, in order to observe Google Maps behaviour.

STAGE 1: GPS-SDR-SIM AND GNSS-SDR

The emulation starts with GPS-SDR-SIM, tool that generates GPS baseband signal data stream. We used it to build up the raw signal, that in a real attack should be transmitted with a Software Defined Radio, such as HackRF, LimeSDR, USRP (online it is possible to find a lot of guides on the most famous SDR devices).

This tool allows us to decide some parameters, the most important are:

- RINEX navigation file, allows the user to post-process the received data to produce a more accurate result
- Sampling frequency
- I/Q data type, set the right number of bits for I/Q modulation
- Spoofed path, file which contains the spoofed coordinates (it accepts ECEF or NMEA GGA format)

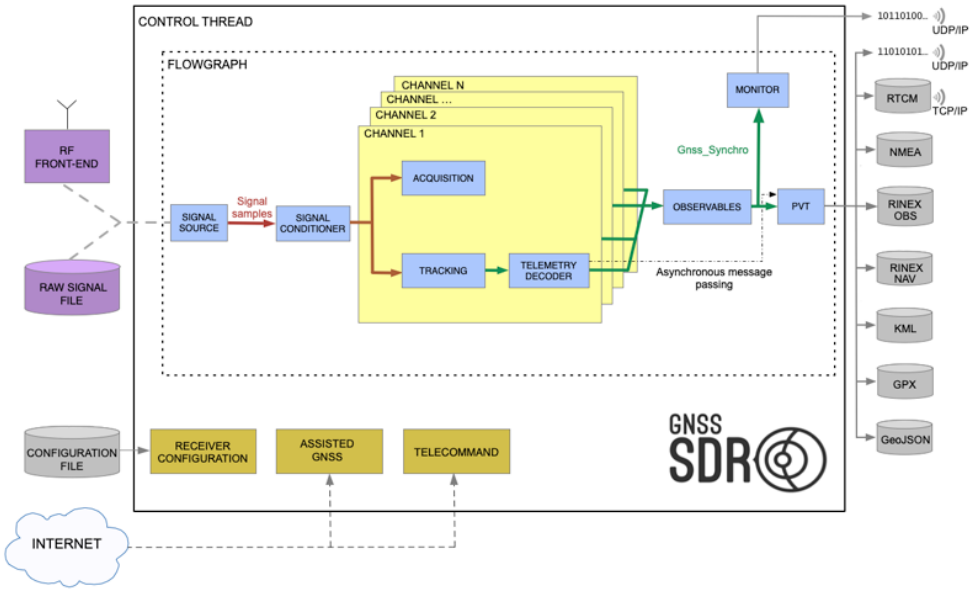
This command generates our signal

```
gps-sdr-sim -e brdc3540.14n -u SpoofedPath.csv -s 4000000
```

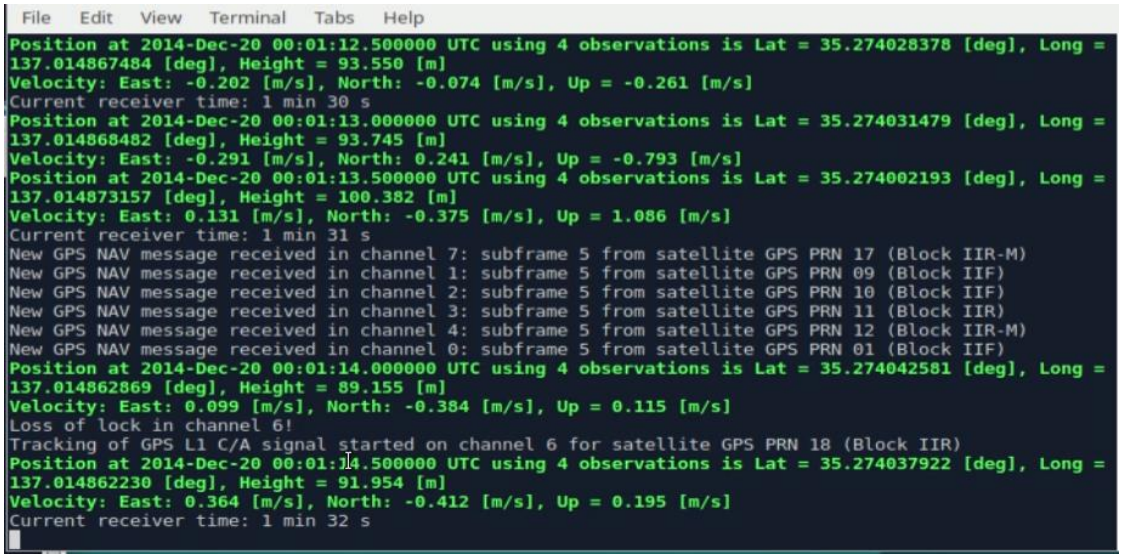
```
Start time = 2014/12/20,00:00:00 (1823:518400)
Duration = 300.0 [sec]
01   75.5  19.0  23842014.1  17.4
02  278.2   8.6  25211959.5  16.6
03   44.8  38.2  22161310.3  11.1
06  298.1  43.4  21779609.4   9.6
09  144.3  29.1  22867978.9  13.9
10  222.8  34.6  22165342.9  11.6
11   98.7   7.0  25049095.4  24.3
12  326.9   5.0  25187588.2  16.5
17  350.8  83.8  20316661.5   7.3
20   47.2  49.4  21274204.2   9.3
23  106.0  35.7  22595053.5  12.2
28  201.9  27.9  22755566.1  13.5
32   43.1   2.2  25691385.0  25.0
Time into run = 300.0
Done!
Process time = 115.5 [sec]
```

The SpoofedPath.csv contains coordinates of the ghost route the victims think to run across, displayed by their navigators. These records must be written in the ECEF format (Earth-Centered Earth-Fixed, online there are a lot of ***Latitude,Longitude,Height*** **to** ***X,Y,Z*** converter, but most of them do not allows stream of coordinates, so we made a python script, the result are calculated referring to the ellipsoid WGS-84).

The next part about finding the right receiver's configuration has been the most difficult one because of lack of documentation about the pair GNS-SDR-SIM & GNSS-SDR. The emulator receiver GNSS-SDR is a modular, very configurable and powerful software, but, for this reason, the configuration is the crucial part to deal with.



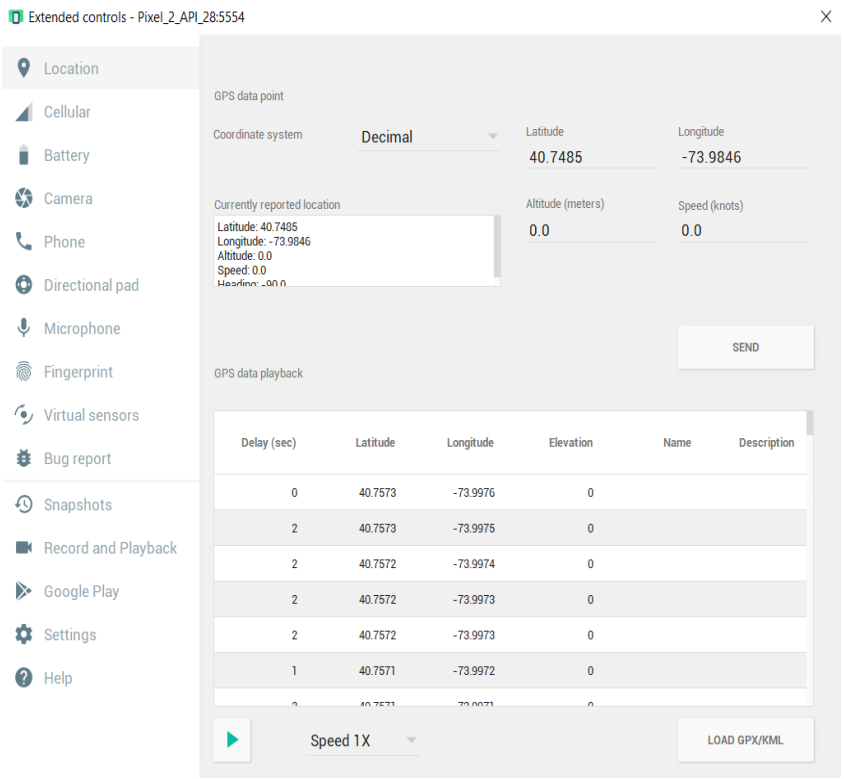
We built our configuration basing on the default (https://gnss-sdr.org/conf/) and adjusting the parameters to be able to correctly read the previous generated raw signal.



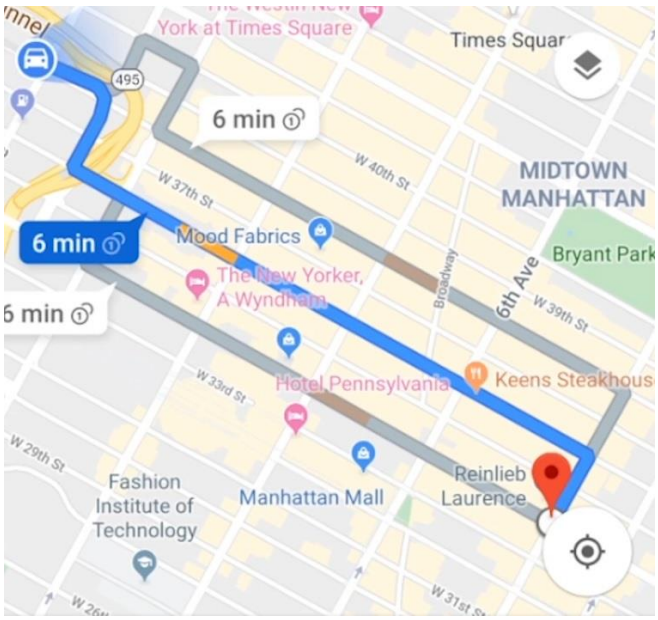
Thanks to these coordinates, saved in a GPX output file, it was possible to emulate the attack on Google Maps.

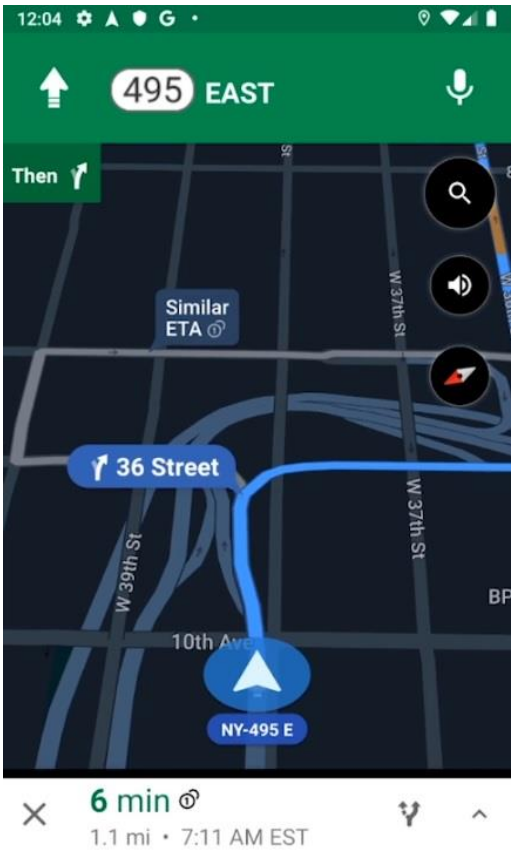
STAGE 2: QEMU AND GOOGLE MAPS

QEMU is an emulator that performs hardware virtualization and that allowed us to start virtual devices running Android 9.0 with Google Maps already installed. In the location settings, it is possible to load a GPX file containing all our location, that is basically the file generated as result in the previous stage.

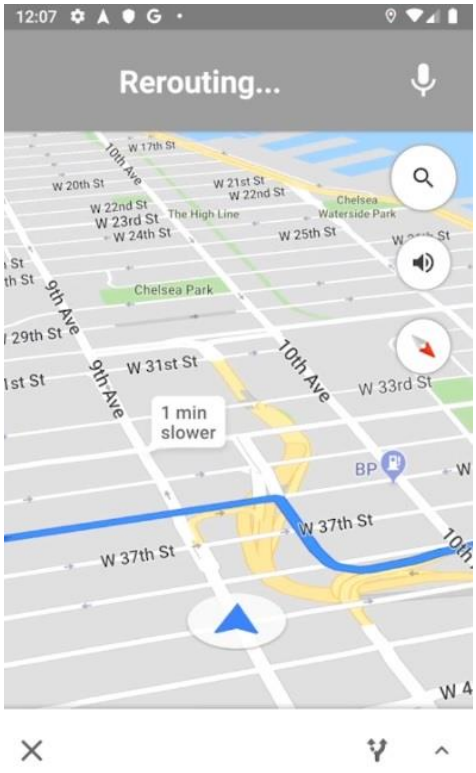


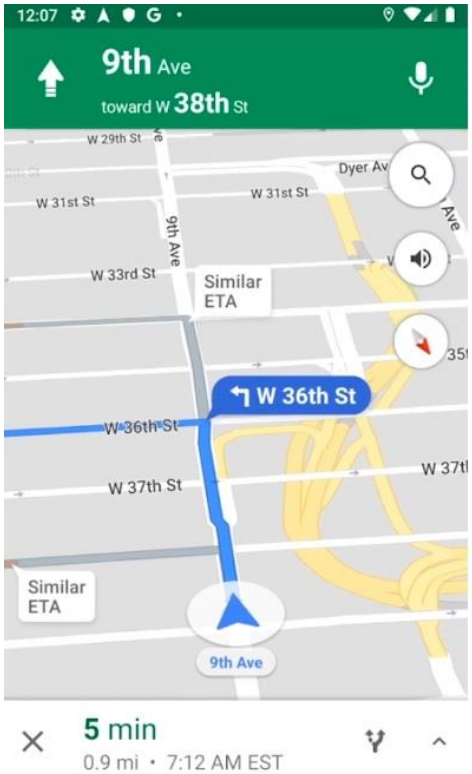
After setting Google Maps to navigate from the current driver location to his destination, we could start the simulation forcing all the locations in the GPX file one after another.





After an initial period during which everything seems to proceed normally (during this phase the attacker should worry about the takeover), the jump to the ghost location is triggered.





It really takes just a bunch of seconds for the app to reroute ($\approx 5s$): the user could not even watch the navigator screen during this short amount of time. After the jump, the navigation proceeds as the driver

was still proceeding to the destination, but in reality he leaves the original route after the first turn: it is now crucial that there is a strict correspondence between the ghost route and the victim route, otherwise the navigation system could give the driver an instruction without any apparent sense, making him understand something wrong is going on. In order to calculate such ghost routes, in the paper a research algorithm is presented, returning the list of all the possible ghost positions and routes evaluated on a given path.

USER STUDY

Since we managed to partially reproduce the hack in a virtual environment, we can confirm that Google Maps behaviour makes it look feasible. The paper documents some user tests that were conducted on actual people to measure the efficiency of the attack, and we now report them here in order to use them in the next section for our observations. Before starting the driving tests, users were asked to two questions:

- *How often do you use GPS navigation services when driving in familiar locations and unfamiliar locations?* Answers lead to 89.25% for unfamiliar ones, 42.2% for familiar ones.
- *What information provided by the navigation service do you primarily rely on during driving?* Answers lead to 68.4% for voice prompt, 57.9% visual elements, 31.6% textual information.

Due to safety implications, the driving tests were conducted in a simulated environment, “Euro Truck Simulator II”: the attack achieved a success rate of 95%. During a final interview, it came out that most users experienced GPS malfunction in real life (95%), and that among 40 participants, only 8 could explain GPS spoofing correctly.

OBSERVATIONS AND CONCLUSIONS

As specified in the previous section, the driving tests show a high successful rate: however, these tests were held in a simulation, and there is not trace in the paper if traffic conditions were emulated too. During our work we could observe a couple of Google Maps curious behaviours that could compromise the success of the attack, both regarding actual traffic considerations, and both not taken in consideration inside the paper.

JUMP REROUTING

Emulating the attack in different time slots during the day, sometimes after a jump an unpredictable new route was calculated due to Google Maps consideration on the actual traffic conditions. This will basically lead the driver to make turns in not predicted points, inconsistent with the positions that will be forced by the attacker, revealing the hack.

PHYSICAL ROUTE INCOMPATIBILITY

Again, during our simulations, it was not that uncommon to be notified by the app about the unusual speed we were keeping on a traffic street: the ghost locations sent by the attacker do not take the traffic factor in consideration, but this can lead to a desynchrony between the ghost and the victim route, increasing the risk of an hack failure.

Both these observations can be solved performing the hack in the absence of traffic. For the first one it can also be considered not to run the attack in cities with dense road networks, but this decreases the performances of the ghost routes research algorithm.

Besides these considerations, we still believe this hack is feasible and dangerous: several frightening applications are listed in the paper, such as ambulances and police cars detouring, victim misguidance for ambush, rubbering or stealing, wrong way entrance on a highways, etc. For this reason, we agree on the conclusions of the paper: the results underline the lack of practical defence mechanism to protect the massive GPS users and GPS-enabled autonomous system, such as location verification, signal authentication, sensor fusion, etc.

EMULATION: BETTER SOFTWARE OR HARDWARE?

This experience made us understand that a software implementation is not always “softer”: online there are a lot of prebuilt configurations for real SRD-card, whereas using GNSS-SDR, we tried a lot of possible setups before reaching a working one. By the way, once found, being able to make work the couple GPS-SDR-SIM and GNSS-SDR allowed local experimentation which resulted much more flexible and quicker than using any other hardware devices.