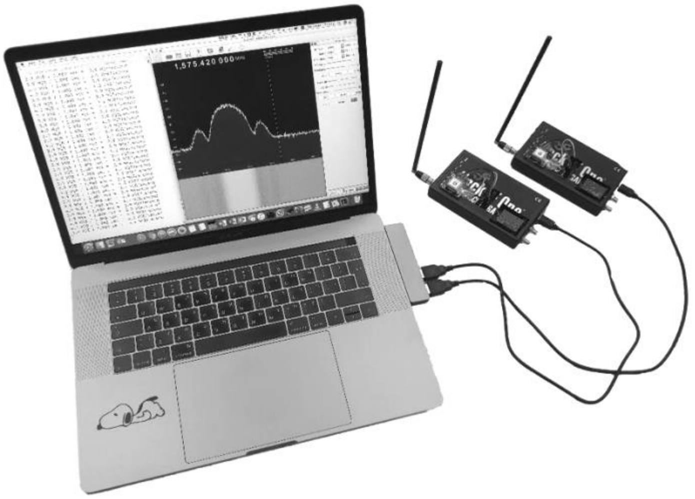
REGULUS

RESILIENCY REPORT

Global Navigation Satellite Systems

Foreword



GNSS spoofing refers to the action of deceiving a GNSS receiver by transmitting GNSS signals from an unauthentic source. The generating and transmitting of falsified GNSS signals at a slightly stronger level than the authentic signals causes the targeted GNSS receiver to accept the fabricated signals, validate them and use them to calculate and report the **wrong position or time**. This represents a risk to many missions and systems that rely upon GNSS data for navigation, timing, localization, sensor fusion, etc. In the past, GNSS spoofing attacks required many resources and expensive equipment and were usually reserved for the military, other government agencies and well-funded organizations. However, in recent years software defined radios (SDR) and open-source GNSS simulators have become widely available to the public. This has led to an increase of the feasibility and the occurrence of non-military GNSS spoofing attacks.

Standard GNSS receivers are merely designed to provide navigation data and typically do not have the ability to detect security breaches, such as spoofing attacks. There are no solutions today for the commercial market that detect and mitigate spoofing attacks. There are some military-grade solutions utilizing controller radiation-pattern antenna (CRPA) technology that, by design, combat jamming attacks, with some modifications to handle spoofing attacks. Another solution for military systems is to disregard satellite navigation data suspected of being spoofed and not use it to calculate the geographic location, however this detection capability is not available for commercial GNSS receivers.

A wide variety of systems rely on GNSS signals both for timing and location. As the spoofing threat is now at the hands of non-military hackers, a solution to identify and mitigate GNSS spoofing attacks is needed.

Contents

1 Introduction

For the past year, Regulus has been developing the Pyramid GNSS technology that enables the detection of spoofed GNSS signals for commercial GNSS receivers. During this time, numerous lab tests and field test were conducted to verify the reliability of the detection technology. Hence, in addition to detecting GNSS spoofing attacks, Regulus has been developing advanced GNSS spoofing capabilities, all using open source hardware and software, to aid in the development of the detection technology and reveal the vulnerabilities of commercial GNSS receivers. The results of these tests and experiments are compiled in this document.

The first part of this report demonstrates the vulnerability of the most popular commercial GNSS receivers as well as high-end GNSS receivers.

The second part of this report demonstrates the performance of the Pyramid GNSS technology in terms of successful detection of spoofing attacks and false positives.

In this report, we refer to "false positive" as a state where no spoofing attack is taking place, but an alert of a spoofing attack is generated. We refer to "spoofing detection" as a state where a known spoofing attack is taking place and an alert of a spoofing attack is generated.

The goal of the technology is to achieve a zero percent false positive rate and a one hundred percent spoofing detection rate.

2 Vulnerabilities

Regulus has developed very robust spoofing capabilities in order to test how commercial receivers cope with such attacks, both in a lab environment and outside in the real world.

2.1 GNSS Receiver Spoofing

A GNSS receiver is a device (usually a single chip and an antenna) that receives RF signals as its input and sends a Position, Velocity and Time (PVT) solution as its output. The three layers are shown in Figure 1.

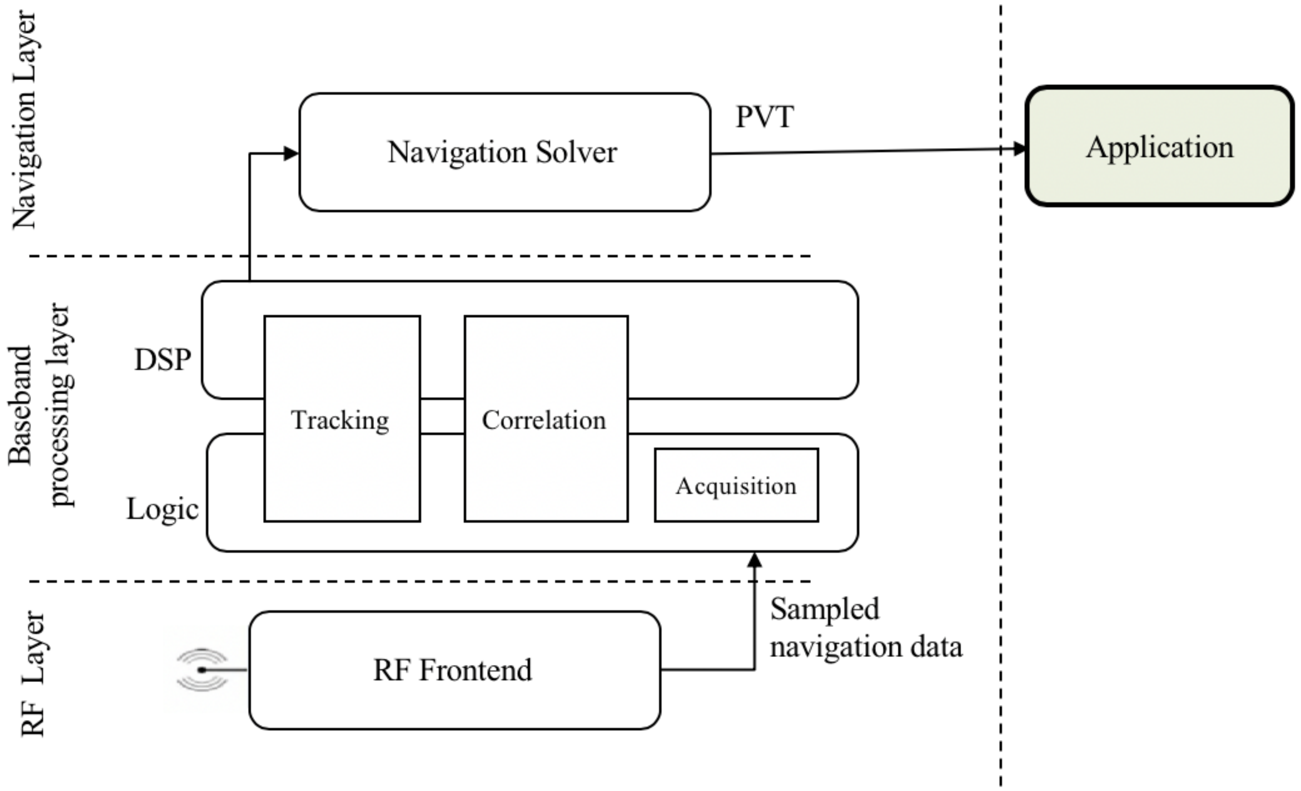


Figure 1: GNSS Architecture

Spoofing a receiver is a process where carefully crafted RF signals are transmitted by an attacker, causing the receiver to report a false PVT solution.

2.2 Sensor Fusion Spoofing

Sensor fusion systems combine data received from different sensors to reach a more comprehensive result and understanding of the environment. Spoofing a system that works with sensor fusion is a process where at least one of the sensors is attacked in a way that will cause the system to report a wrong PVT solution. Sensor fusion systems rely on data coming from multiple sensors that aid each other to provide a more reliable PVT solution. In most of these systems, the GNSS receiver has the highest weight in the algorithm. Accordingly, spoofing the receiver is a good approach into spoofing the whole system.

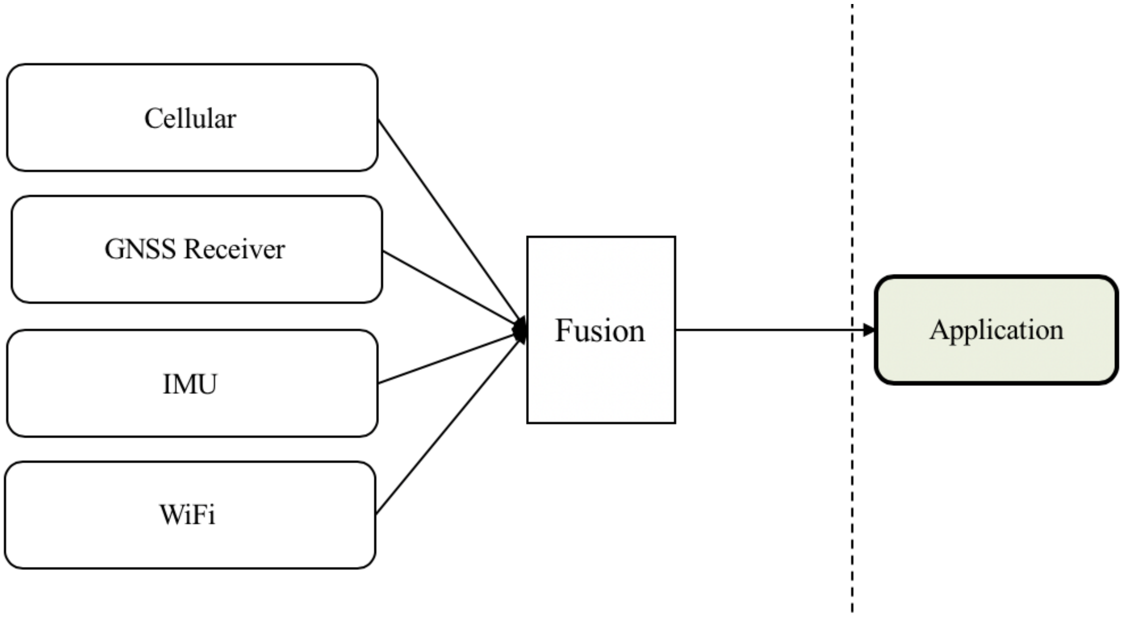


Figure 2: Sensor Fusion Architecture

2.3 Spoofing Setup

It should be noted that high-end RF equipment (e.g., $5,000 spectrum analyzers and $100,000 signal generators) can be used as a spoofing setup. However, due to the high cost of the equipment, this type of setup is very rare in the hands of an ordinary hacker. For this reason, the spoofing setup used in the presented experiments includes only low-cost hardware (e.g., HackRF One and BladeRF) and open source software.

Hardware:

1. Laptop
2. HackRF +TCXO
3. Linearly polarized omni directional antenna
4. Low cost GNSS receiver as a PPS source

Software:

1. Modified version of the software-defined GNSS signal simulator (github gps-sdr-sim)

2. Low cost software defined radio (github hackrf)

Example for generating a signal:

```
./gps-sdr-sim -e brdc -b 8 -l 47.25,8.30,450 -t 2017/11/06,10:56:03 -s 2000000 -d 150 -o gpssim.bin
```

Example for transmitting a signal:

```
./hackrf_transfer -t gpssim.bin -f 1575420000 -s 2000000 -a 1 -x 5
```

Example for recording a signal:

```
./hackrf_transfer -r rfdata.bin -l 40 -g 30 -s 2000000 -S 1048576 -f 1575420000 -p 1 -a 1
```

2.4 Attack Classification

Since most sensor fusion systems providing PVT rely heavily on the GNSS receiver, GNSS receiver spoofing and sensor fusion spoofing attacks are simply referred to as spoofing attacks.

There are different types and degrees of spoofing attacks, ranging from a simple low power replay attack to very sophisticated real-time attacks. The range between the spoofer and the target is also an important factor when classifying these attacks, although it is mostly a factor of cost (amplifier and antenna).

When classifying spoofing attacks, we also must distinguish between two scenarios: indoor spoofing, where no real GNSS signals are available, and outdoor spoofing where some or all GNSS signals are available. In both cases, the most important parameters for classifying an attack, are *ease of execution* and *detection difficulty*. Ease of execution refers to the effort (i.e., hardware and software tools) required to execute an attack. Detection difficulty refers to the techniques required to be implemented in a GNSS receiver (not a sensor fusion system) to detect an attack successfully.

Indoor spoofing attacks are considered easier to perform since there is no real GNSS signal with which to compete. In most cases this is true, but not always. As an example, a target receiver is a part of a car or mobile phone where sensor fusion with an odometer, WiFi, Cellular, Bluetooth and IMU can aid in positioning and detect anomalies. But for these systems, GNSS is the primary data source for PVT data and when it is available, whether legitimately or as a spoofed signal, the phone and car will use it.

Outdoor attacks are harder to perform since, in most cases, a GNSS receiver is already locked-on to a real signal. In this case, the easiest way for an attacker to break the lock is to use a GNSS jammer. Another way is to transmit at a much higher RF power until the attacked receiver switches over to the spoofed signal. If an attacker starts an attack before a GNSS receiver is powered on, it will be much easier for the attacker to make the receiver lock on to the spoofed signal.

Class 0: This class includes the meaconing and replay attacks. Meaconing is the interception and rebroadcast of GNSS signals. These signals are rebroadcasted on the received frequency, typically, with power higher than the original signal to confuse the attacked receiver, causing it to calculate a wrong PVT solution.

These kinds of attacks can be performed using a *hackrf_transfer* to both record the RF signals in the form of I/Q data stored on a computer, and when needed, re-transmit the recorded file using the same hardware.

Detecting this class of attacks is relatively easy and can be done by looking for time changes that are discontinuous or location discrepancies in short intervals.

Class 1: This class includes attacks with a pre-defined scenario, which can be static or moving. The scenarios are generated and stored on a computer in the form of I/Q data files. When required, these files are transmitted.

These kinds of attacks can be done with *gps-sdr-sim* to generate the scenarios and *hackrf_transfer* to transmit the generated signals.

Detecting this class of attacks is relatively hard since the spoofer's scenario can be very realistic, with the time being very close to the actual time and correct position at the beginning of the scenario, which makes a robust anomaly detection hard to implement. In addition, multi-constellation receivers can be used to compare locations, assuming they are not jammed.

Class 1M: Similar to the previous class but with an added capability of spoofing multiple constellations. This class of attacks requires extra hardware (more than one SDR) and more software to generate Galileo, GLONASS and BeiDou signals.

Detecting this class of attacks is harder than detecting Class 1, since no jamming is required by the attacker.

Class 2: This class includes real-time attacks in which the spoofer is perfectly synced in time and phase to the authentic signal. The attacker also uses an up-to-date ephemeris data to be perfectly aligned with all the navigation information that the real satellites are transmitting. These attacks create a smooth takeover with absolutely no time discrepancies and no location discrepancies. The attacker can define a scenario and generate the required I/Q data in real-time. The attacker can also alter the generated location during the attack.

These kinds of attacks are done by using a slightly modified gps-sdr-sim version coupled with a slightly modified *hackrf_transfer* function to generate and transmit a scenario in real-time. Here, the attacker must use HW synchronization to be perfectly aligned in time with the authentic signal.

Detecting this class of attacks is extremely hard since the spoofer's scenario starts with the correct time and correct position to get a seamless takeover, and after a while starts drifting and offsetting the position.

Class 2M: Similar to the previous class but with an added capability of spoofing multiple constellations. This class of attacks requires extra hardware (more than one SDR), all synced with the real GNSS signals and more software to generate Galileo, GLONASS and BeiDou signals.

Detecting this class of attacks is harder than detecting Class 2 since no jamming is required by the attacker.

Class	Ease of Execution	Realtime
0	Easy	No
1	Easy	No
1M	Medium	No
2	Medium	Yes
2M	Hard	Yes

Table 1: Spoofing Attacks Summary

2.5 Vulnerabilities

2.5.1 Targets

The first target group that was tested are standalone receivers. All these receivers use only GNSS signals to provide a PVT solution (unlike mobile phones or cars for example, that can also use an odometer, IMU, WiFi and Cellular connectivity). The second target group that was tested were mobile phones. These devices are considered to be a sensor fusion system where the end user receives a location based on several inputs, mainly the GNSS, WiFi and Cellular. Unlike standalone receivers, the final PVT solution provided to the user is a combination of all sensors, e.g., provide a position inside a building where no GNSS signals are available. The third group that was tested were cars. They are also considered to be a sensor fusion system where the end user receives a location based on several inputs, mainly the GNSS, odometer, steering wheel angle and IMU. Unlike standalone receivers, the final PVT solution provided to the user is a combination of all sensors, e.g., provide location in a tunnel or underground garages.

Standalone Receivers

The number of OEM GNSS receiver manufacturers is much higher than the number of GNSS chip manufacturers. However, most OEMs use the most popular GNSS chipsets for their mass-market receivers (mainly UBX by u-Blox, MT3339 by MediaTek and SiRFstar by Qualcomm).

In this report, the most popular mass market receivers were selected along with high end receivers, to construct a comprehensive picture of the vulnerabilities of GNSS receivers in the commercial market.

Manufacturer	Model
GlobalTop	PA6C/GTPA010
u-Blox	NEO-6M
u-Blox	NEO-7M
GlobalSat	G-Star IV
STM	Teseo-LIV3F
u-Blox	NEO-M8
Furuno	GN-87
Javad	TRH-G2

Table 2: List of Target Receivers

Mobile Phones

The phones that were selected for the tests are a good representation of both the low-end and high-end market of devices. They are all equipped with a multi-constellation GNSS receiver.

Manufacturer	Model
Apple	iPhone XS
Samsung	Galaxy Prime Pro
Huawei	Mate 10 Pro
Xiaomi	Mi8

Table 3: List of Target Mobile Phones

Cars

The cars that were selected for the tests all have built-in navigation systems and some autonomous capabilities (e.g., adaptive cruise control, super cruise, ADAS). It is not known what kind of sensor fusion is used, but it is highly likely that the odometer and steering wheel angles are used as inputs to the GNSS receiver as well as an IMU.

Manufacturer	Model
Mercedes	CLS 400D
BMW	BM 318A
Cadillac	CT6
Tesla	S

Table 4: List of Target Cars

2.5.2 Indoor Tests

Standalone Receivers and Mobile Phones

The indoor tests were performed inside a lab, where no external GNSS signals are available. The tests were performed on each of the targets listed in Table 2 and Table 3.

Spoofer Setup:

1. HackRF + TXCO.
2. Low gain, linear polarization, omni directional antenna.
3. Laptop running the Regulus spoofer software (using *hackrf_transfer* and *gps-sdr-sim*).

Test Procedure:

1. Position the target 50cm away from the spoofer.
2. In case of a standalone receiver, connect the receiver to a laptop to view and log the navigation messages (i.e., lat, lon, alt, time).
3. In case of a mobile phone, open the Google Maps app and look for the "blue dot."
4. Power on the target.
5. Initiate the spoofing attack (t_1).
6. In case of a standalone receiver, record the time it takes to get a 3D fix ($t - t_1$).
7. In case of a mobile phone, record the time it takes the "blue dot" to change its position ($t - t_1$).

Cars

The indoor tests were performed inside an underground parking garage, where no external GNSS signals are available. The tests were performed on each of the targets listed in Table 4.

The spoofer setup and procedure were similar to the one used to test the standalone receivers and phones. In this case, the spoofer was transmitting from inside the car.

2.5.3 Indoor Tests Results

Manufacturer	Model	Successfully Spoofed	Spoofed After [sec]
GlobalTop	PA6C/GTPA010	Yes	6
u-Blox	NEO-6M	Yes	19
u-Blox	NEO-7M	Yes	6
GlobalSat	G-Star IV	Yes	4
STM	Teseo-LIV3F	Yes	37
u-Blox	NEO-M8	Yes	7
Furuno	GN-87	Yes	5
Javad	TRH-G2	Yes	30
Manufacturer	**Model**	**Successfully Spoofed**	**Spoofed After [sec]**
Apple	iPhone XS	Yes	5
Samsung	Galaxy Prime Pro	Yes	5
Huawei	Mate 10 Pro	Yes	5
Xiaomi	Mi8	Yes	5
Manufacturer	**Model**	**Successfully Spoofed**	**Spoofed After [sec]**
Mercedes	CLS 400D	Yes	5
BMW	BM 318A	Yes	5
Cadillac	CT6	Yes	5
Tesla	S	Yes	5

Table 5: Indoor Test Results

2.5.4 Outdoor Tests

Standalone Receivers and Mobile Phones

The outdoor tests were performed in an area with a clear line of sight to the sky. The tests were performed on each of the targets listed in Table 2 and Table 3.

Spoofer Setup:

1. HackRF + TXCO.
2. GNSS receiver as a 1PPS source.
3. Low gain, linear polarization, omni directional antenna.
4. Laptop running the Regulus spoofer software (using *hackrf_transfer* and *gps-sdr-sim*).

Test Procedure:

1. Position the target 50cm away from the spoofer.
2. In case of a standalone receiver, connect the receiver to a laptop to view and log the navigation messages (i.e., lat, lon, alt, time).
3. In case of a mobile phone, open the Google Maps app and look for the “blue dot”.
4. In case of a car, open the built-in navigation system and look for the “arrow icon”.
5. Perform scenario A and scenario B.

In scenario A, the spoofing attack is initiated after the target has locked on a real GNSS signal:

1. Power on the target.
2. Wait for the target to obtain a fix on the current position.
3. Wait for 1 minute to obtain a solid lock.
4. Initiate the spoofing attack (t_1).
5. In case of a standalone receiver, record the time it takes ($t - t_1$) to get a 3D fix on the spoofed position.
6. In case of a mobile phone, record the time it takes ($t - t_1$) the “blue dot” to change its position.

In scenario B, the spoofing attack is initiated before the target is powered on. Since a mobile phone is always on, it was not a part of this scenario:

1. Initiate the spoofing attack.
2. Power on the target (t_1).
3. Record the time it takes ($t - t_1$) to get a 3D fix on the spoofed position.

Cars

The outdoor tests were performed, in an area with a clear line of sight to the sky. The tests were performed on each target listed in Table 4.

The spoofer setup and procedure were similar to the one used to test the standalone receivers and phones. In this case, the spoofer was transmitting from inside the car.

2.5.5 Outdoor Tests Results

Table 6 shows the results of the spoofing tests in scenario A. It is measured in terms of how long it took to spoof the receiver and if the attack was successful.

Manufacturer	Model	Successfully Spoofed	Spoofed After [sec]
GlobalTop	PA6C/GTPA010	Yes	33
u-Blox	NEO-6M	Yes	5
u-Blox	NEO-7M	Yes	7
GlobalSat	G-Star IV	Yes	30
STM	Teseo-LIV3F	Yes	31
u-Blox	NEO-M8	Yes	21
Furuno	GN-87	Yes	23
Javad	TRH-G2	Yes	40
Manufacturer	**Model**	**Successfully Spoofed**	**Spoofed After [sec]**
Apple	iPhone XS	Yes	30
Samsung	Galaxy Prime Pro	Yes	10
Huawei	Mate 10 Pro	Yes	50
Xiaomi	Mi8	Yes	20

Table 6: Results of Outdoor Spoofing, Scenario A

Table 7 shows the results of the spoofing tests in scenario B. It is measured in terms of how long it took to spoof the receiver and if the attack was successful.

Manufacturer	Model	Successfully Spoofed	Spoofed After [sec]
GlobalTop	PA6C/GTPA010	Yes	85
u-Blox	NEO-6M	Yes	11
u-Blox	NEO-7M	Yes	4
GlobalSat	G-Star IV	Yes	10
STM	Teseo-LIV3F	Yes	30
u-Blox	NEO-M8	Yes	6
Furuno	GN-87	Yes	24
Javad	TRH-G2	Yes	20
Manufacturer	**Model**	**Successfully Spoofed**	**Spoofed After [sec]**
Mercedes	CLS 400D	Yes	20
BMW	BM 318A	Yes	25
Cadillac	CT6	Yes	33
Tesla	S	Yes	29

Table 7: Results of Outdoor Spoofing, Scenario B

3 Spoofing Effects

As part of the spoofing tests, it is worth mentioning the effects of a spoofing attack and what it may cause.

3.1 Standalone Receivers

The obvious and immediate effect on the receiver is that it reports a wrong position and/or time. The way a *system* is affected by the use of wrong data really depends on the system. In the next two sections, a short discussion is presented describing the finding of how a mobile phone and cars are affected by the spoofing attack.

3.2 Mobile Phones

All location-based services are not useable. A few examples:

- Display the wrong position on the map.
- Unable to plan or follow a route.
- Unable to use a navigation app like Waze, Google Maps and Apple Maps.
- Unable to use ride hailing apps like Uber, DiDi and Lyft.
- Unable to use fitness tracking apps like Endomondo and Strava.
- Find My iPhone does not work.
- Wrong photo geo-tagging.

After a spoofing attack has stopped, in many cases, it takes a very long time for the phone to recover from the attack and display the real position.

This has major privacy implications where a user can be “placed” in a location that he is not.

3.3 Cars

The effects that were observed during the testing on a car can be divided into two categories: safety and non-safety.

Safety:

- Adaptive cruise control tried to accelerate to 100 km/h in an urban area where the speed limit was 30 km/h. Driver had to intervene and slam on the brakes.
- Adaptive cruise control unexpectedly, and for no reason, slowed down to 50 km/h on a main road where the speed limit was 100 km/h.

- ADAS systems failed to slow down before an intersection.
- ADAS system activated the breaks unexpectedly and for no reason on the main road while driving 100 km/h, thinking the car was approaching an intersection.
- The height of the car’s suspension was changed unexpectedly and for no reason while driving.
- SOS feature reports the wrong position to dispatch.
- Confusing and distracting navigation cues while trying to follow a planned route.

Non-safety:

- The car’s built-in navigation system displays the wrong position on the map.
- Unable to plan or follow a route.
- Unable to activate adaptive cruise control.
- GPS-based alarm services do not work.

4 Pyramid Performance

The first part of the report clearly indicated that commercial GNSS receivers and mobile phones are extremely vulnerable to GNSS spoofing attacks.

The following section of the report shows the performance of the Pyramid GNSS technology under the same set of attacks in terms of detection accuracy and false positive rate.

4.1 False Positives

The goal of this test set is to provide insight regarding a false positive rate i.e., how many times the Pyramid GNSS receiver reported a spoofing attack while no spoofing attack was taking place. The tests are divided into three types: stationary, driving mode and pedestrian mode. Each test set was performed in different locations and different times of day. During these tests, the Pyramid GNSS receiver was exclusively exposed to authentic GNSS signals and monitored regarding its output.

4.1.1 Stationary Mode

This test set was conducted multiple times in different locations (see Google Earth screen shots below). At each location, the Pyramid GNSS receiver was placed on a steady surface without any movement throughout the test.

Test ID SA001, 32.780644,35.001335:

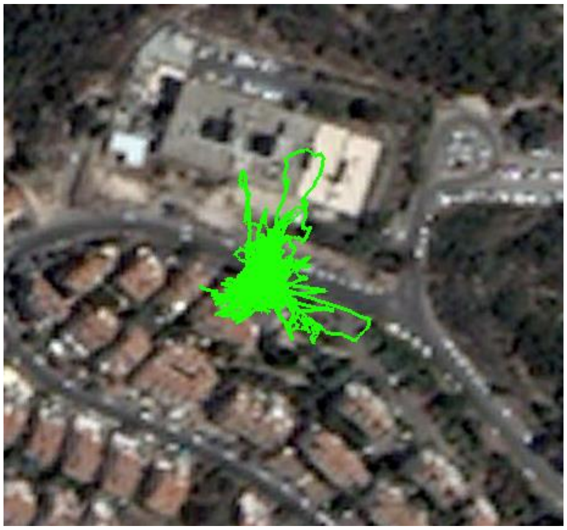


Figure 3: Test SA001 Data Tracks

Test ID SA002, 32.780644,35.001335:

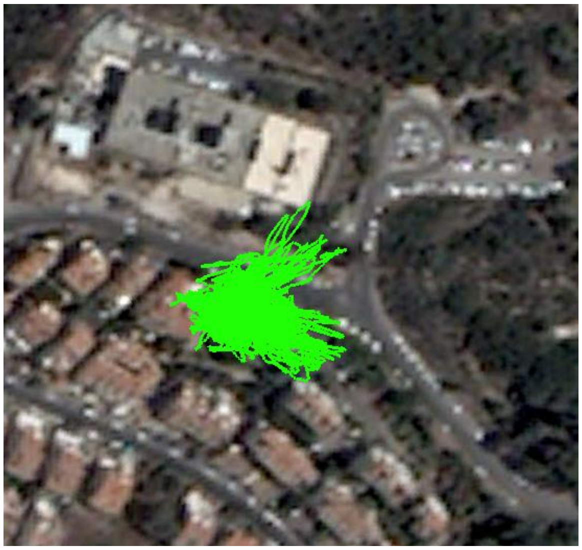


Figure 4: Test SA002 Data Tracks

Test ID SB001, 32.981383,35.083402:

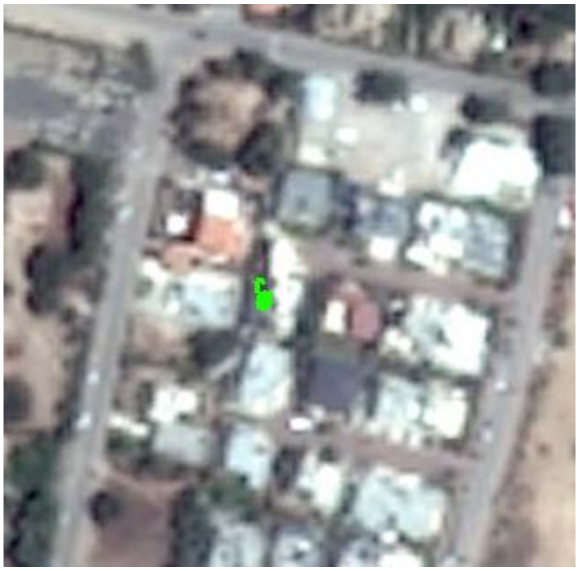


Figure 5: Test SB001 Data Tracks

Test ID SB002, 32.981383,35.083402:

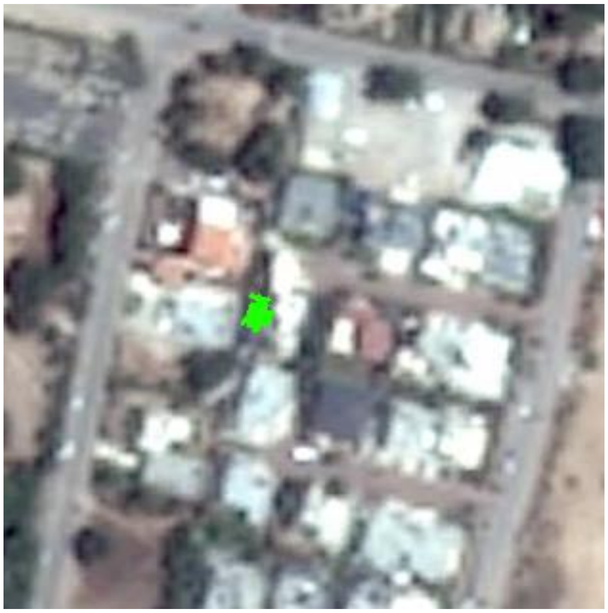


Figure 6: Test SB002 Data Tracks

4.1.2 Driving Mode

This test set was conducted along three different routes (see Google Earth screen shots below). Along each route, the Pyramid GNSS receiver was placed on the dashboard of a car while driving at an average speed along normal daily commute routes.

Test ID RA001, 186km:

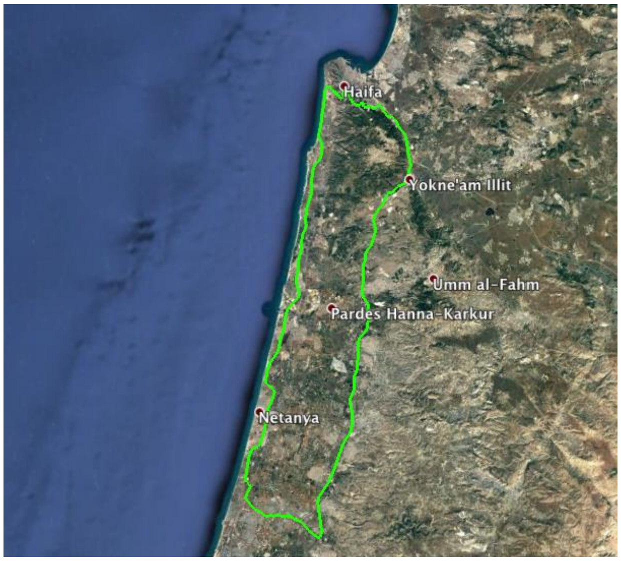


Figure 7: Test RA001 Data Tracks

Test ID RB001, 7.14km:

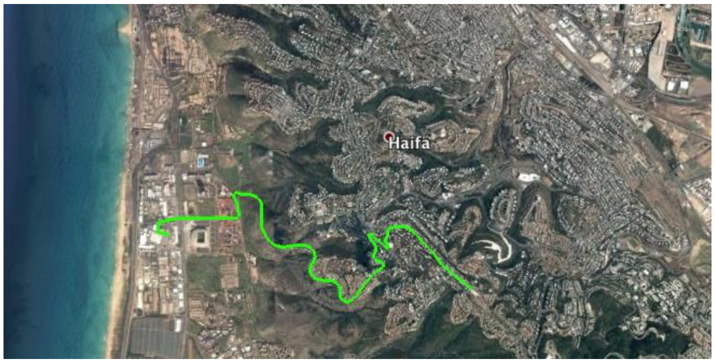


Figure 8: Test RB001 Data Tracks

Test ID RC001, 167km:

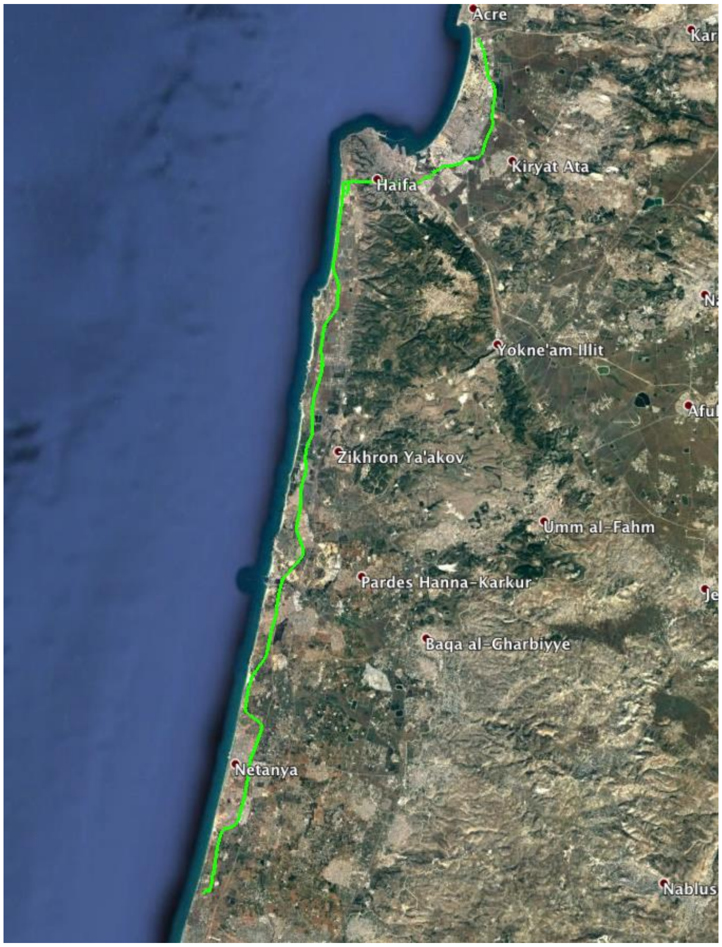


Figure 9: Test RC001 Data Tracks

4.1.3 Pedestrian Mode

In this test, the Pyramid GNSS receiver was carried by hand while walking around at a normal pace.

Test ID PA001, 32.782997,34.960387:

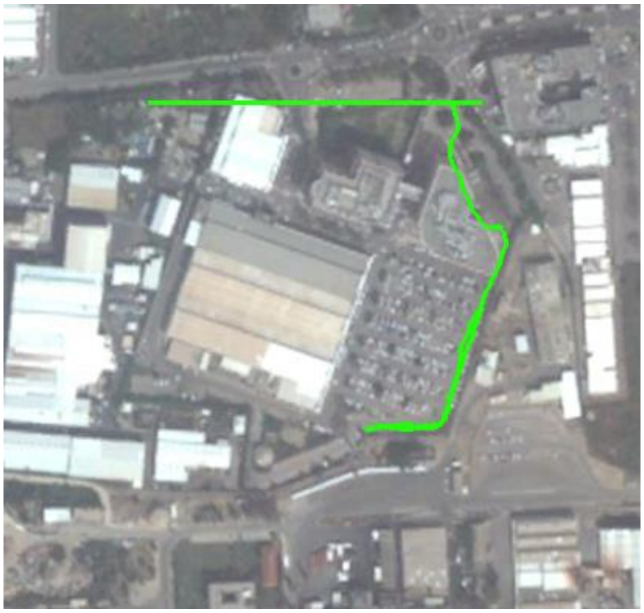


Figure 10: Test PA001 Data Tracks

Test ID PA002, 32.782997,34.960387:

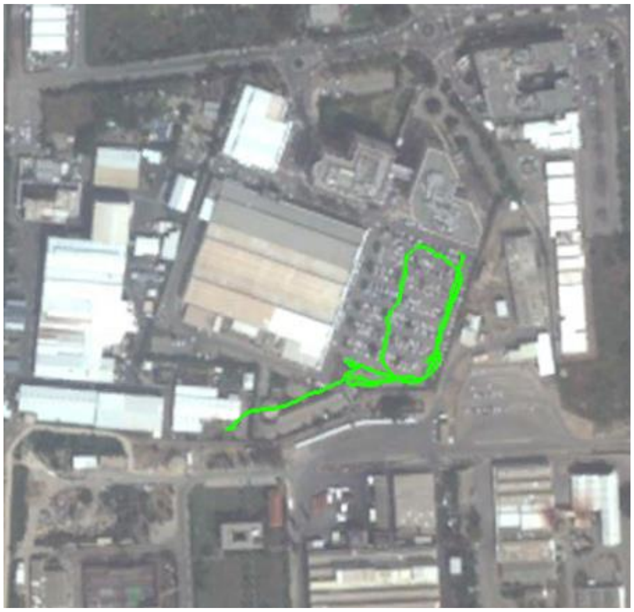


Figure 11: Test PA002 Data Tracks

4.1.4 Results

Test ID	Start Date	Start Time	Duration	Data Points	Spoofing Detections
SA001	21.12.2018	07:13	11:21	40862	0
SA002	21.12.2018	20:29	12:48	46116	0
SB001	31.12.2018	15:41	12:34	45210	0
SB002	01.01.2019	22:37	6:12	22314	0
RA001	22.12.2018	13:42	6:43	20608	0
RB001	23.12.2018	05:59	0:20	1209	0
RC001	02.01.2019	05:40	7:49	28082	0
PA001	08.01.2019	11:57	0:19	1127	0
PA002	06.01.2019	17:49	0:11	675	0

Table 8: Results of False Positive Tests

During the tests, a total of 206,203 data points (nearly 60 hours) were collected with zero spoofing detections. At no point did the Pyramid GNSS receiver falsely identify a spoofing attack.

4.2 Detection Accuracy

The goal of this test set is to provide insight regarding detection rate, i.e., how many times the Pyramid GNSS receiver reported a spoofing attack while an actual spoofing attack was taking place.

4.2.1 Indoor Tests

The indoor tests were performed inside a lab, where no external GNSS signals are available. The tests were performed on the Pyramid GNSS receiver.

Spoofer Setup:

1. HackRF + TXCO.
2. Low gain, linear polarization, omni directional antenna.
3. Laptop running the Regulus spoofer software (using *hackrf_transfer* and *gps-sdr-sim*).

Test Procedure:

1. Position Pyramid GNSS receiver 50cm away from the spoofer.
2. Initiate the spoofing attack (t_1).
3. Record the time it takes for a detection alert to be displayed ($t - t_1$).
4. Repeat steps 1-3 for 8 test instances.

4.2.2 Results of Indoor Tests

Test Instance	Detection Time [sec]	Spoofing Detected
1	18	Yes
2	4	Yes
3	3	Yes
4	13	Yes
5	7	Yes
6	11	Yes
7	8	Yes
8	15	Yes

Table 9: Results of Indoor Detection Accuracy Tests

4.2.3 Outdoor Tests

The outdoor tests were performed in an area with a clear line of sight to the sky. The tests were performed on the Pyramid GNSS receiver.

Spoofer Setup:

1. HackRF + TXCO.
2. GNSS receiver as a 1PPS source
3. Low gain, linear polarization, omni directional antenna.
4. Laptop running the Regulus spoofer software (using *hackrf_transfer* and *gps-sdr-sim*)

Test Procedure:

1. Position the receiver 200cm away from the spoofer.
2. Perform scenario A and scenario B.

In scenario A, the spoofing attack is initiated after the receiver is locked-on a real GNSS signal:

1. Power on the receiver.
2. Wait for the receiver to obtain a fix on the current position.
3. Wait for 1 minute to obtain a solid lock.
4. Initiate the spoofing attack (t_1).
5. Record the time it takes for a detection alert to be displayed ($t - t_1$).

In scenario B, the spoofing attack is initiated before the receiver is powered on:

1. Initiate the spoofing attack.
2. Power on the receiver (t_1).
3. Record the time it takes ($t - t_1$) to get a 3D fix on the spoofed position.

Repeat each scenario for 8 test instances.

4.2.4 Results of Outdoor Tests

Scenario	Test Instance	Detection Time [sec]	Spoofing Detected
A	1	3	Yes
	2	4	Yes
	3	4	Yes
	4	3	Yes
	5	2	Yes
	6	2	Yes
	7	2	Yes
	8	2	Yes
B	1	10	Yes
	2	10	Yes
	3	7	Yes
	4	10	Yes
	5	9	Yes
	6	7	Yes
	7	8	Yes
	8	7	Yes

Table 10: Results of Outdoor Detection Accuracy Tests

5 Conclusion

Spoofing attacks on GNSS receivers should be considered a serious threat. There is sufficient motivation for this type of illicit hacking and performing a spoofing attack is feasible and not very difficult. As such, it is anticipated that many research activities will be conducted on increasing the security of GNSS receivers against spoofing attacks.

Until now, GNSS spoofing was not perceived as a safety issue. In this report, several critical issues were observed while driving a car that uses GNSS for advanced features.

In this report, different spoofing scenarios were described and the vulnerabilities of many GNSS receivers were demonstrated. The Regulus Pyramid GNSS has proven to be very effective in detecting GNSS spoofing attacks, while being extremely reliable with a very low false positive rate.

This report is a work in progress: additional GNSS receivers are being tested for vulnerabilities, spoofing attacks are being tested in different scenarios and location, and the Pyramid GNSS receiver is under continual testing, evaluation, and development.

6 Solution

Regulus Cyber is solving the GNSS spoofing attacks threat using Pyramid GNSS. Pyramid GNSS is solving the GNSS hacking threat affecting the automotive, aviation, maritime, and mobile industries with a unique technology applicable both as a fortified GNSS Receiver, capable of detecting spoofing attacks, and at the chip level, allowing mobile phones, cars, and IoT devices to receive GNSS spoofing protection for the first time. The company was able to miniaturize its technology into an industry-leading form factor that provides customers with more flexibility with integration.

As the dependency on GNSS and Satellite Signals grows, so does the need for safety and security. While real attacks are expanding, anti-spoofing solutions remain a luxury that only high-end, defense markets can afford. The Regulus Pyramid GNSS brings a real revolution to the resilient GNSS eco-system, allowing GNSS spoofing detection and mitigation at the PCB level and later on at the chip level. While current solutions are big, heavy and expensive, Pyramid GNSS offers industry standard size and price. For the first time ever, vast industries such as automotive, aviation, maritime, and mobile phones can defend themselves against this sophisticated emerging threat, at an affordable price and relevant size, power consumption, and weight.

About Regulus Cyber

Regulus Cyber is the first company to develop an end-to-end sensor cybersecurity solution. Regulus was founded by a team of engineers and entrepreneurs that have extensive knowledge and experience in creating security solutions for the military, homeland security, and commercial projects and they bring that expertise to the relevant commercial markets. Founded in 2016, Regulus is based in Haifa, Israel and is backed by Sierra Ventures, Canaan Partners Israel, the Technion, and F2 Capital. For more information, visit **www.regulus.com**

7 Index

7.1 Abbreviations

PVT	Position, Velocity and Time
GPS	Global Positioning System
GNSS	Global Navigation Satellite Systems
RNSS	Regional Navigation Satellite Systems
SDR	Software Defined Radio
CRPA	Controller Radiation Pattern Antenna/Array
TCXO	Temperature Compensated Crystal Oscillator
PPS	Pulse Per Second
RF	Radio Frequency
GNU	GNU's Not Unix
I/Q	In phase and Quadrature
IMU	Inertial Measurement Unit
LBS	Location Based Services

7.2 List of Figures

7.3 List of Tables

7.4 Revision History

Revision	Date	Status / Comments
-	01/01/2019	Internal
A	10/01/2019	Internal
B	27/01/2019	Released
C	05/02/2019	Internal
D	18/03/2019	Added cars that were spoofed