GPS Spoofing on Android

By: Elena Bucciarelli, Colin Quinn, Reilly Parent

CS 457: Wireless Security

Introduction

GPS spoofing is a technique in altering the calculated location of a device. GPS, or global positioning system, is a technology that uses groups of satellites to triangulate position. These satellites are arranged in constellations that orbit Earth in a consistently distanced position in order to provide the most accurate calculations possible. Multiple constellations can also be used in conjunction with one another so that more points of information can give a more accurate triangulation calculation. The Global Navigation Satellite Systems (GNSS) orbit at 12,550 miles above Earth’s surface, meaning that as the signals travel from satellite to receiver, the signal is rather weak. The signal weakness in combination with the lack of encryption or any sort of authentication leads to a very vulnerable technology. In addition to these vulnerabilities, GPS sensors and transmitters are rather accessible to most people, meaning that almost anyone with the motivation to can perform some sort of attack on their personal or other’s locations.

GPS Spoofing

Many people are interested in GPS spoofing for many different reasons. Anyone from the President of the United States to a coffee shop hacker to a teenager trying to watch region-bound Netflix shows has a reason to know about GPS spoofing. Another very common use of spoofing is for testing location based applications and finding any potential errors in development processes. This technique is a way to mask or change the perceived location of a device, without having to physically move that device.

There are a few different techniques to performing GPS spoofing, one being useful for attacking others and another being useful for hiding the location of a personal device. The far more difficult type is attacking other’s GPS receivers. In order to perform this type of attack, the attacker must recreate the signals that are being sent from the satellite constellations. The device that is spoofing the satellite signals must be able to emulate multiple different signals. It is able to do so by using the unencrypted Coarse Acquisition (C/A) codes.These C/A codes are open to the public which leads to relatively simple data extraction and duplication. An attacker in this case is able to create a device that emulates sending these C/A codes as a fake satellite. If the emulation is good enough, then any GPS receiver will now think that the emulated device is the satellite constellation that gives it the most accurate location. Meanwhile the attacker is able to set that location to anywhere that they see fit, or simply make it a random latitude and longitude. This is typically most useful in cases such as the military, especially the Navy as they rely heavily on GPS locations to navigate large bodies of water.

The more simple method of GPS spoofing has far more personal use cases. This method utilizes a testing feature in most phones or computers that allows the user to choose their location based on a map. Using this, applications that rely heavily on location can be faked into thinking the device is at a different physical location. Things such as Google Maps, Netflix, social media applications, online stores, video games, and any other technology that utilizes location are vulnerable to this attack. These are typically less impactful in negative ways, but simply allow the user to access the internet from different parts of the world. For example, Netflix is known to region-lock some shows and using a GPS spoofer like this would allow for anyone to connect from the region that has their desired show and watch it as normal. Similarly, Pokemon GO has region specific Pokemon, so using this method of GPS spoofing would let anyone trick their device into being that region and would then be able to capture any type of Pokemon from around the world.

Methodology

We implemented our GPS spoofing attack via an Android application. Utilizing Google Cloud's Maps API in combination with Android's *ALLOW_MOCK_LOCATION* debug permission, we were successful in creating a barebone application that allows the user to select the perceived location of the device. Below is a code snippet that creates a fake location and sends it to the system's GPS. The android class being used to do this is *LocationManager*, which you can read more about here.

```
Location loc = new Location(prov); //prov is the GPS service

loc.setLatitude(lat); // lat and long obtained from Google Maps API
loc.setLongitude(lon);
loc.setAccuracy(1);
loc.setTime(System.currentTimeMillis());
loc.setElapsedRealtimeNanos(SystemClock.elapsedRealtime());

locationManager.setTestProviderLocation(prov, loc);
```

This code block runs as a foreground service. The service continues to run even if the main app *activity* (or screen) gets destroyed or closed. This allows the location of the device to be mocked even without the app running. A notification is shown on the phone's notification center if their location is being spoofed - this is a requirement to run code as a foreground service. A logging screen was also implemented to help debug the app throughout development.

The App

We decided to develop an Android application called “Spoofer” that allows the user to select anywhere on the map and switch the device’s location to the newly selected location. Shown in Figure 1 is the application upon opening. The location selected is depicted by a large red pin and it can be placed anywhere on the map. Once a location is selected, Android will notify the system via Android Intents shown in Figure 2. Further information can be displayed on the log screen, which gives data such as latitude, longitude, and acceleration (which is unknown due to the pin location being stationary). Logs for this pinned location can be seen in Figure 3. Also shown in Figure 4 is the application altering the location on Snapchat as we can see that it has moved the user to Antarctica.

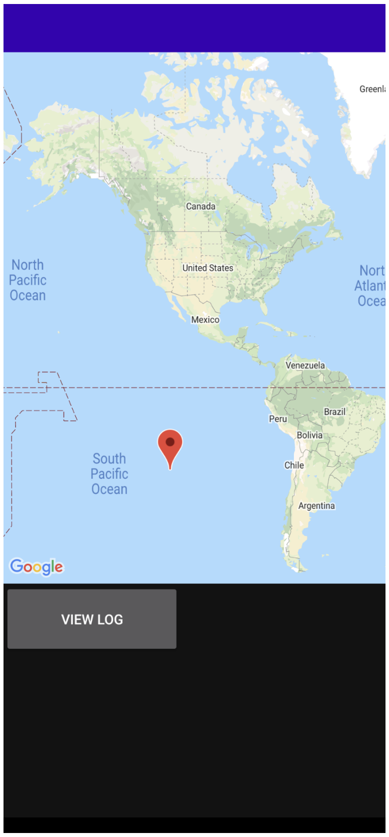
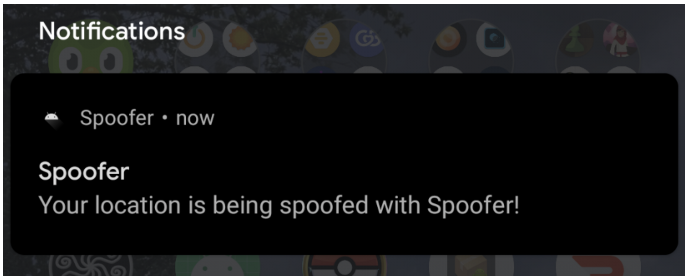


Figure 1 - Main page of Spoofer

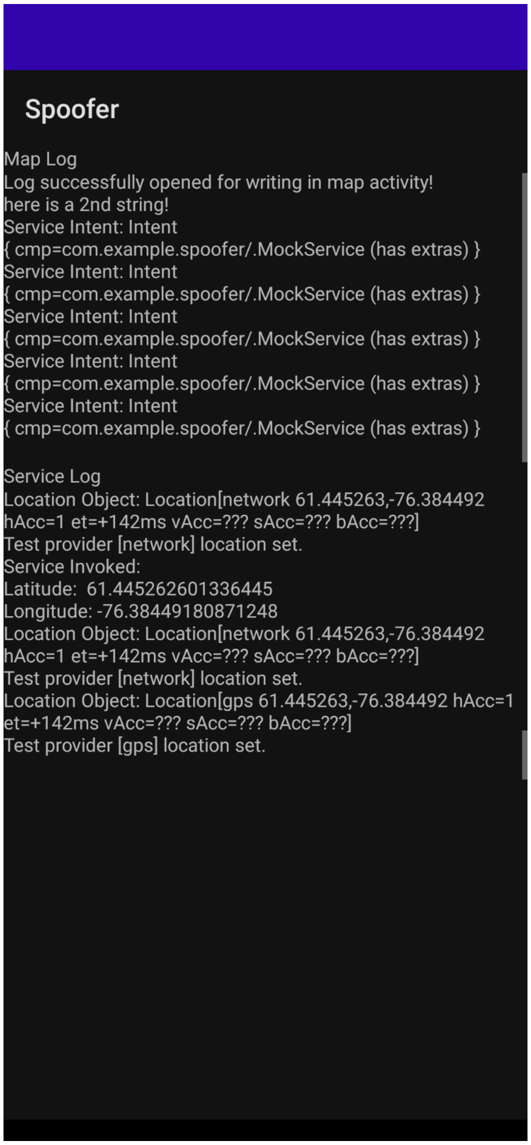


Notifications

Spoofer • now

Spoofer
Your location is being spoofed with Spoofer!

Figure 2 - Spoofer foreground service notification



Spoofer

```
Map Log
Log successfully opened for writing in map activity!
here is a 2nd string!
Service Intent: Intent
{ cmp=com.example.spoofer/.MockService (has extras) }
Service Intent: Intent
{ cmp=com.example.spoofer/.MockService (has extras) }
Service Intent: Intent
{ cmp=com.example.spoofer/.MockService (has extras) }
Service Intent: Intent
{ cmp=com.example.spoofer/.MockService (has extras) }
Service Intent: Intent
{ cmp=com.example.spoofer/.MockService (has extras) }

Service Log
Location Object: Location[network 61.445263,-76.384492
hAcc=1 et=+142ms vAcc=??? sAcc=??? bAcc=???]
Test provider [network] location set.
Service Invoked:
Latitude:  61.445262601336445
Longitude: -76.38449180871248
Location Object: Location[network 61.445263,-76.384492
hAcc=1 et=+142ms vAcc=??? sAcc=??? bAcc=???]
Test provider [network] location set.
Location Object: Location[gps 61.445263,-76.384492 hAcc=1
et=+142ms vAcc=??? sAcc=??? bAcc=???]
Test provider [gps] location set.
```

Figure 3 - Spoofer log screen

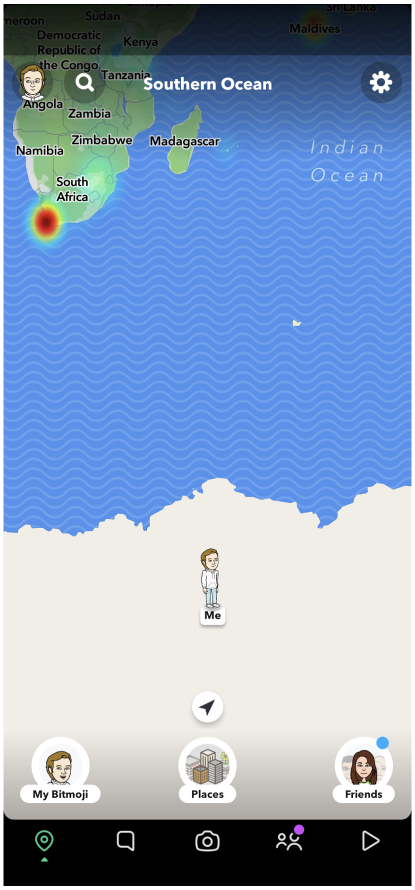


Figure 4 - A fake location from Spoofer shown in Snapchat

Exploits

The exploits that this application enables are very useful to many average people. Social media very often tracks location data from images that are posted and allows the user to display their location with each post. Streaming services such as Netflix and Hulu often lock certain content for any given region, and performing a GPS spoofing attack lets anyone move to any region and consume any content they want. Video games such as Pokemon GO are also well known to region-lock specific Pokemon to only be available in specific areas. This attack allows anyone to trick their device into being in any location so that they can catch these area specific items.These use cases are mostly for the average person, however, GPS spoofing can be used to do less innocent things as well. Spoofing location is useful when trying to hide your identity online, similar to using a VPN to anonymize internet traffic.

Defense Against GPS Spoofing

Defending against GPS spoofing is not a simple task. Some GPS receivers have a technology called Selective Availability Anti-Spoofing Module, which allows the receiver to track an encrypted code back to the device that sent it. From there it is possible to verify that a real satellite has sent the data that is being read. This solution however, is currently only available to the Department of Defense. As a civilian, multi-constellation receivers with built-in inertial measurement units (IMU) are the most reliable at preventing any spoofing attacks. Collecting information from multiple constellations allows for the certainty of location to override any potential spoofers. In combination with an IMU that is able to calculate information like velocity, roll, pitch, yaw, and current position, the user will have the best opportunity at preventing any GPS spoofing attempts.

The attack utilizing a physical device to emulate satellites has a proven solution, while the personal device technique of GPS spoofing does not. Applications that spoof location on devices like Android rely heavily on the debug setting that enables mock location. This reliance is the turning point of a functioning GPS spoofer, and without the setting enabled, the application will fail. Therefore it is possible to discover which applications have these settings enabled, and block service until those settings are disabled. This can be done via the function named *Location.isFromMockProvider()* found in Android API 18. This function is not always fully functional as some applications do not label their location as being mocked.

Conclusion

Overall, GPS spoofing is a very useful technique in altering the location of any physical device. There are numerous use cases varying from high profile individuals using it to hide their location, to Netflix watchers trying to watch a region-restricted series. The most widely used variation of GPS spoofing is very simple to implement and enables users to do many different things. There are also not many reliable ways to detect a mocked location or prevent others from spoofing locations.

GitHub

If you want to see our whole codebase, it is available on GitHub:
https://github.com/SuchFNS/GPSspoofing

Sources

Ball, B. (2020, December 28). Why gps spoofing is a problem (and what to do about it). NextNav. Retrieved September 13, 2021, from https://nextnav.com/gps-spoofing/.

Understanding the difference between Anti-Spoofing And anti-jamming. NovAtel. (2013). Retrieved September 13, 2021, from https://novatel.com/tech-talk/velocity-magazine/velocity-2013/understanding-the-difference-between-anti-spoofing-and-anti-jamming.