Protection against fingerprinting with Generic Sensor API

Today's devices contain various sensors for reading information about the device's position, state, and environment. Such equipment is typical for mobile devices like cellphones, tablets, or laptops that often include sensors for obtaining geolocation or device orientation data. Another example is a smartwatch that could monitor the heartbeat rate of the wearer, or a car with a tire pressure sensor, etc. While the benefits of having sensors are undisputed, allowing websites to access their readings represents a considerable danger.

Generic Sensor API

JavaScript's Generic sensor API provides a unified way for accessing these sensors and reading data. The physical (hardware) sensor instances are called device sensors, while platform sensors represent interfaces over which the user agent can interact with the device sensors and read data. Platform sensors are represented by a series of JS classes. The base class Sensor cannot be used directly but provides essential properties, event handlers, and methods for its subclasses. Those represent concrete sensors like Accelerometer, Magnetometer, or Gyroscope.

Browser Support

The API is currently implemented, or partially implemented, in Chrome, Edge, and Opera browsers. For Android devices, the support exists in Chrome for Android, Opera for Android, and various Chromium-based browsers like Samsung Mobile or Kiwi Browser. The concrete support for individual classes depends on the browser type and version. Some features are considered experimental and do only work when browser flags like #enable-experimental-web-platform-features or #enable-generic-sensor-extra-classes are enabled.

Sensor Types

Some sensors are characterized by their implementation, e.g. a Gyroscope or Magnetometer. Those are called low-level sensors. Sensors that are named after their readings, not their implementation, are called high-level sensors. For instance, the GeolocationSensor may read data from the GPS chip, Wifi networks, cellular network triangulation, or their combination. Using a combination of low-level sensor readings is called sensor fusion. An example is the AbsoluteOrientaionSensor that uses data from the Accelerometer, Gyroscope, and Magnetometer low-level sensors.

Threats

The risk of using Generic Sensor API calls for device fingerprinting is mentioned within the W3C Candidate Recommendation Draft, 29 July 2021. Documented threats include manufacturing imperfections and differences that are unique to the concrete model of the device and can be used for fingerprinting.

Wrapping of Sensor Data

Timestamps

We discovered a loophole in the Sensor.timestamp attribute. The value describes when the last Sensor.onreading event occurred, in millisecond precision. We observed the time origin is not the time of browsing context creation but the last boot time of the device. Exposing such information is dangerous as it allows to fingerprint the user easily. It is unlikely that two different devices will boot at exactly the same time.

The behavior was with the Magnetometer sensor on the following devices:

- Samsung Galaxy S21 Ultra; Android 11, kernel 5.4.6-215566388-abG99BXXU3AUE1, Build/RP1A.200720.012.G998BXXU3AUE1, Chrome 94.0.4606.71 and Kiwi (Chromium) 94.0.4606.56
- Xiaomi Redmi Note 5; Android 9, kernel 4.4.156-perf+, Build/9 PKQ1.180901.001, Chrome 94.0.4606.71

Our new wrapper thus protects device by changing the time origin to the browsing context creation time, whereas the timestamp should still uniquely identify the reading.

Magnetometer

Magnetometer measures strength and direction of the magnetic field at device's location. The interface offers sensor readings using three properties: x, y, and z. Each returns a number that describes the magnetic field aroud the particular axis. The numbers have a double precision and can be positive or negative, depending on the orientation of the field. The total strength of the magnetic field (M) can be calculated as M = sqrt(x^2 + z^2 + y^2). The unit is in microtesla (μT).

Fingerprinting

The Earth's magnetic field ranges between approximately 25 and 65 μT. Concrete values depend on location, altitude, weather, and other factors. Yet, there are characteristics of the field for different places on Earth. The common model used for their description is the International Geomagnetic Reference Field (IGRF). While the magnetic field changes over time, the changes are slow: There is a decrease of 5% every 100 years. Therefore, for the given latitude, longitude, and altitude, the average strength of the field should be stable. Can one determine the device's location based on the data from the Magnetometer sensor? Not exactly. The measured values are influenced by the interference with other electrical devices, isolation (buildings, vehicles), the weather, and other factors. Moreover, the field is not unique - similar fields can be measured in different places on Earth.

What, however, can be determined is the orientation of the device. In the case of a stationary (non-moving) device, the measured values can serve as a

fingerprint. As we experimentally examined, it is also possible to distinguish whether the device is moving or when its environment changes. When a person with a cellphone enters a car or an elevator, the metal barrier serves as isolation, and the strength of the field gets lower rapidly (e.g., from 60µT outside to 27µT inside). A cellphone lying on a case of a running computer can produce values over 100µT, especially if it is near the power supply unit. Either way, the for a single device at the same location in the same environment, the avarege strength of the magnetic field should be stable.

While we consider it unlikely that someone determines the precise location of the device from the Mangetometer values, its data can be used for fingerprinting. For instance, it can be determined wheter the device is moving or not. In case of a stationary device, we can make a fingerprint from the device's orientation. Another fingerprintable value is the average total strength of the field, which should remain stable if the device is at the same position and in the same environment.

Wrapping

To protect the device, we are wrapping the x, y, z getters of the Magnetometer.prototype object. Instead of using the original data, we return artificially generated values that look like actual sensor readings.

At every moment, our wrapper stores information about the previous reading. Each rewrapped getter first checks the timestamp value of the sensor object. If there is no difference from the previous reading's timestamp, the wrapper returns the last measured value. Otherwise, it provides a new fake reading.

We designed our fake field generator to fulfill the following properties:

- The randomness of the generator should be high enough to prevent attackers from deducing the sensor values.
- Multiple scripts from the same website that access readings with the same timestamp must get the same results. And thus:
- The readings are deterministic - e.g., for a given website and time, we must be able to say what values to return.

For every "random" toss-up, we use the Mulberry32 PRNG that is seeded with a value generated from the domainHash which ensures deterministic behavior for the given website. First, we choose the desired total strength M of the magnetic field at our simulated location. This is a pseudo-random number from 25 to 60 µT, like on the Earth. In the current implementation, we simulate a stationary device with a pseudo-randomly drawn orientation. Therefore, we choose the orientation of the device by generating a number from -1 to 1 for each axis. Those values we call baseX, baseY, and baseZ. By modifying the above-shown formula, we calculate the multiplier that needs to be applied to the base values to get the desired field. The calculation is done as follows: mult = (M * sqrt(baseX^2 + baseY^2 + baseZ^2) / (baseX^2 + baseY^2 +

baseZ^2)) Now, we know that for axis x, the value should fluctuate around baseX * mult, etc.

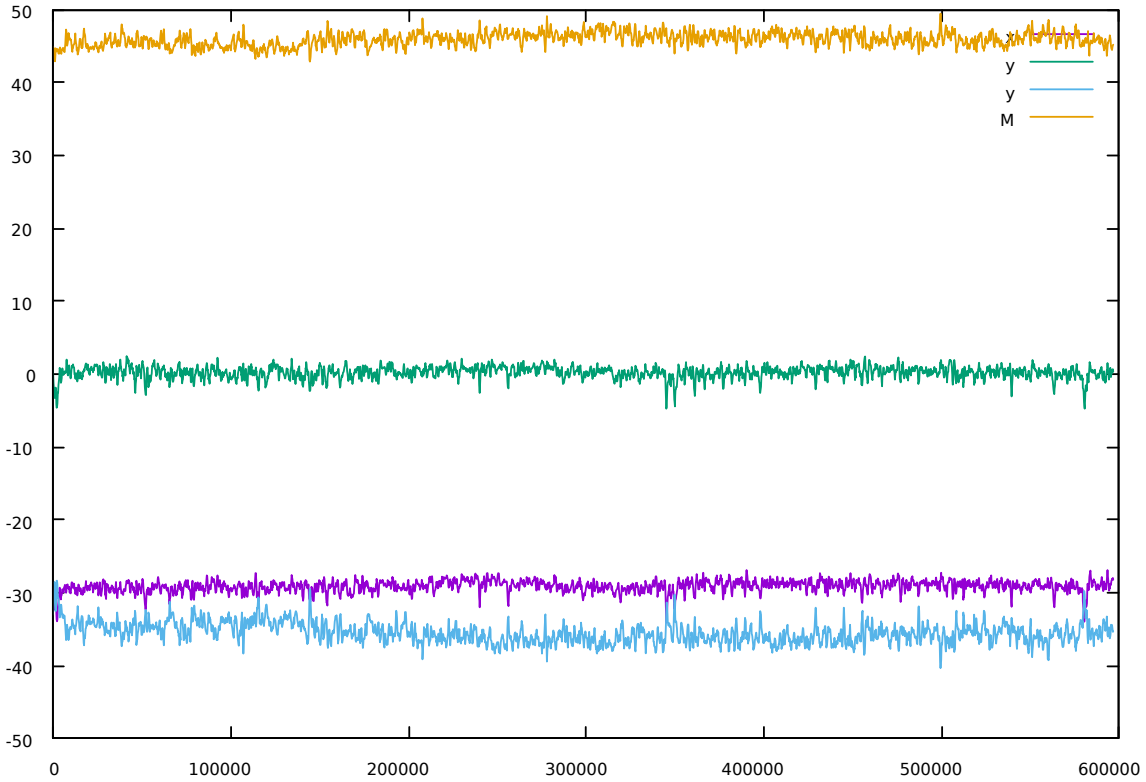
How much the field changes over time is specified by the fluctuation factor from (0;1] that can also be configured. For instance, 0.2 means that the magnetic field on the axis may change from the base value by 20% in both positive and negative way.

The fluctuation is simulated by using a series of sine functions for each axis. Each sine has a unique amplitude, phase shift, and period. The number of sines per axis is chosen pseudorandomly based on the wrapper settings. For initial experiments, we used around 20 to 30 sines for each axis. The optimal configuration is in question. More sines give less predictable results, but also increase the computing complexity that could have a negative impact on the browser's performance.

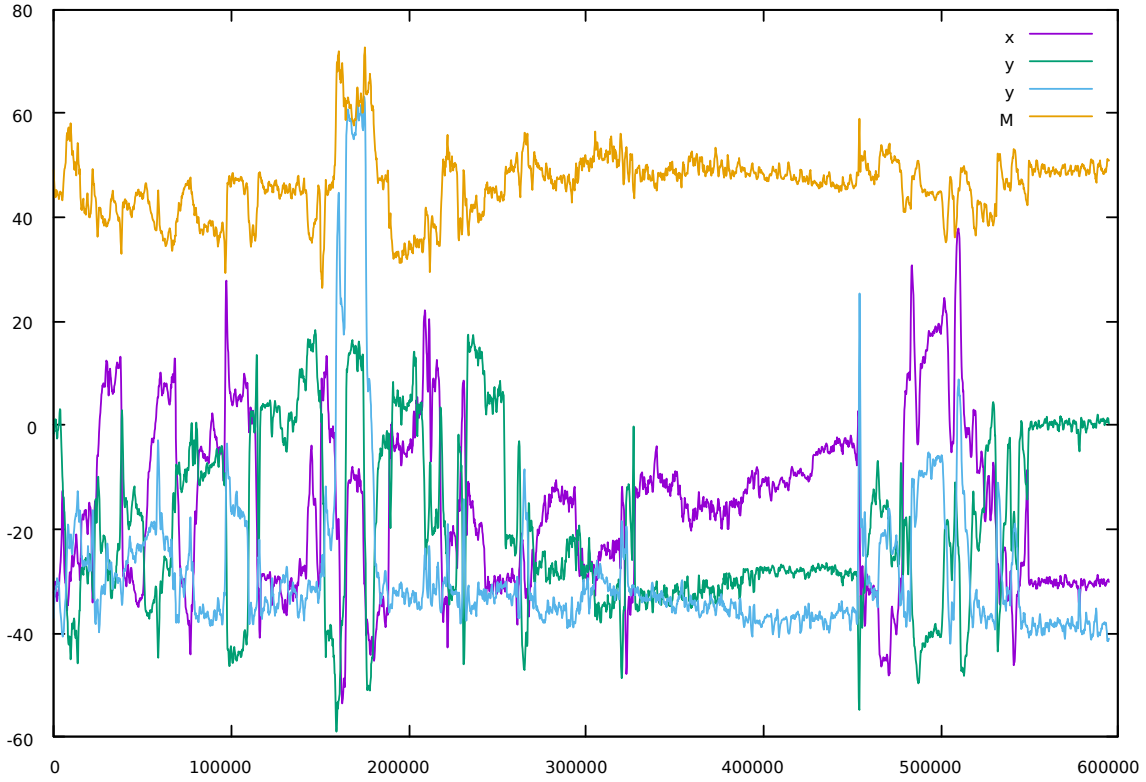
For the given timestamp t, we make the sum of all sine values at the point x=t. The result is then shifted over the y-axis by adding base[X|Y|Z] * multiplier to the sum. The initial configuration of the fake field generator was chosen intuitively to resemble the results of the real measurements. Currently, the generator uses at least one sine with the period around 100 μs (with 10% tolerance), which seems to be the minimum sampling rate obtainable using the API on mobile devices. Then, at least one sine around 1 s, around 10 s, 1 minute and 1 hour. When more than 5 sines are used, the cycle repeats using modulo 5 and creates a new sine with the period around 100 μs, but this time the tolerance is 20%. The same follows for seconds, tens of seconds, minutes, hours. The tolerance grows every 5 sines. For 11+ sines, the tolerance is 30% up to the maximum (currently 50%). The amplitude of each sine is chosen pseudo-randomly based on the fluctuation factor described above. The phase shift of each sine is also pseudo-random number from [0;2π).

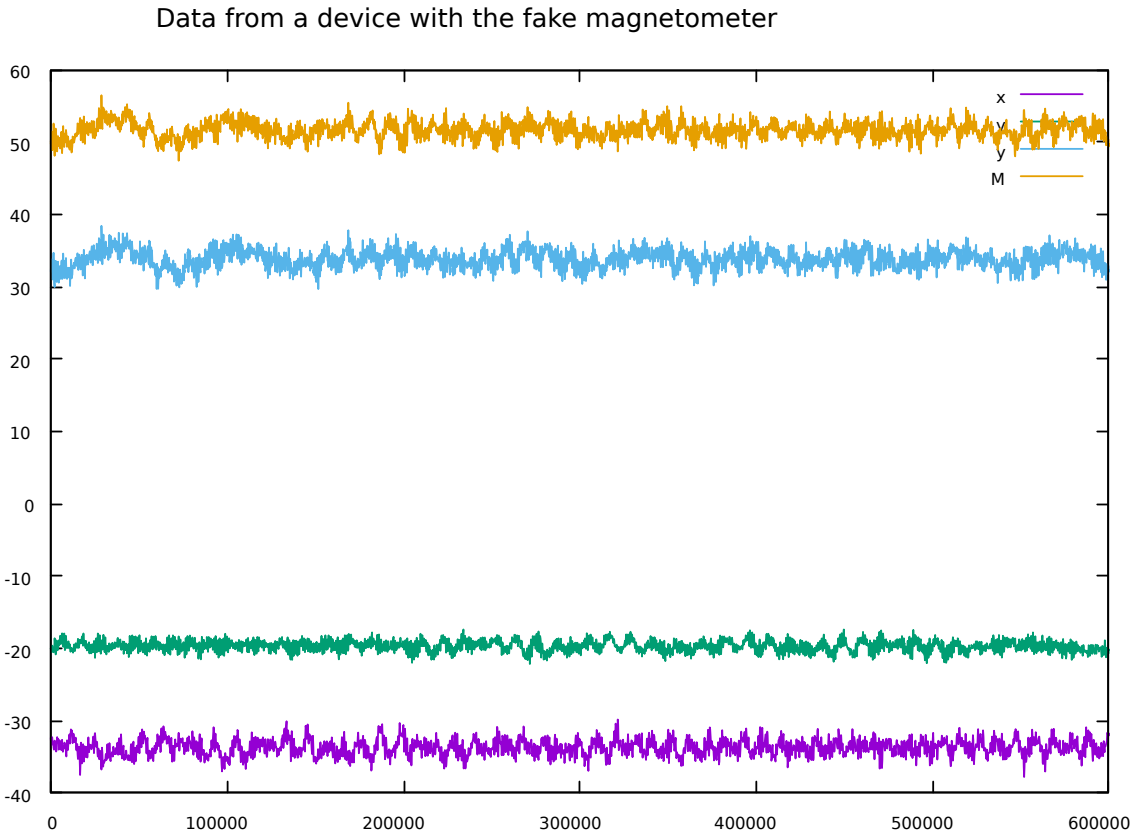
Based on the results, this heuristic returns belivable values that look like actual sensor readings. Nevertheless, the generator uses a series of constants, whose optimal values should be a subject of future research and improvements. Perphaps, a correlation analysis with real mesurements could help in the future. Figures below show the values of x, y, z, and the total strength M measured within 10 minutes on a: 1) Stationary device, 2) Moving device, and 3) Device with the fake wrapped magnetometer.

Magnetometer data from a stationary device



Magnetometer data from a moving device




Data from a device with the fake magnetometer
60
50
40
30
20
10
0
-10
-20
-30
-40
0
100000
200000
300000
400000
500000
600000
x
y
M