The Choice of a Third Authentication Factor

Mark Sitkowski
Design Simulation Systems Ltd
http://www.designsim.com.au

Multi-factor authentication is a good servant, and a bad master.

Experience has proved that, given the choice of convenience or security, most people are far more likely to opt for convenience, if the implementation of the security is such, that it either causes delays in accessing services, or requires the excessive use of brain power.

The widespread use of mobile devices has allowed the major banks to get their users to grudgingly accept the inconvenience of having a second factor thrust upon them, in the form of the SMS-borne second factor.

Other organisations have perpetuated the myth that biometrics is the way of the future, and encouraged their users to delay the queue at Target by scanning fingers, retinas and colonscopy patterns at the checkout.

Our contention is that, as usual, there is a better way, which we have adopted.

As we showed in an earlier paper, https://www.linkedin.com/pulse/choice-second-authentication-factor-mark-sitkowski, there is a better, non-invasive and unobtrusive way of implementing a second authentication factor.
To save the reader the effort of reading the paper, here is a summary of the way we implement it.

We derive a unique signature of the device, without querying it, and without the installation of cookies or client software. The signature is a function of

- The hardware device type, (as in Android, iPhone, Dell laptop, IBM 370 etc)
- The device version number.
- The CPU type and version
- The CPU clock rate
- The operating system used by the device, and its version
- The type and version of the device's graphics adapter
- The browser type used by the client, and its version number (as in Firefox, Opera, Safari etc)

The main criterion driving our choice of a third authentication factor, was that it should strengthen and, in turn, be strengthened by the second factor, described in that paper.

Most importantly, however, it should be totally transparent to the user, and must not impede his access to services.

Since we are already using the device signature as a second factor, it makes sense to use the location of the device as the third.

All browsers are capable of returning geolocation coordinates, but these are so grossly inaccurate, and refer exclusively to the location of the ISP's data centre, that they are useless to us.
Mobile devices, however, are capable of returning latitude and longitude figures, which are accurate to about three feet, or 1.1 metres, so are eminently suitable for our purpose.

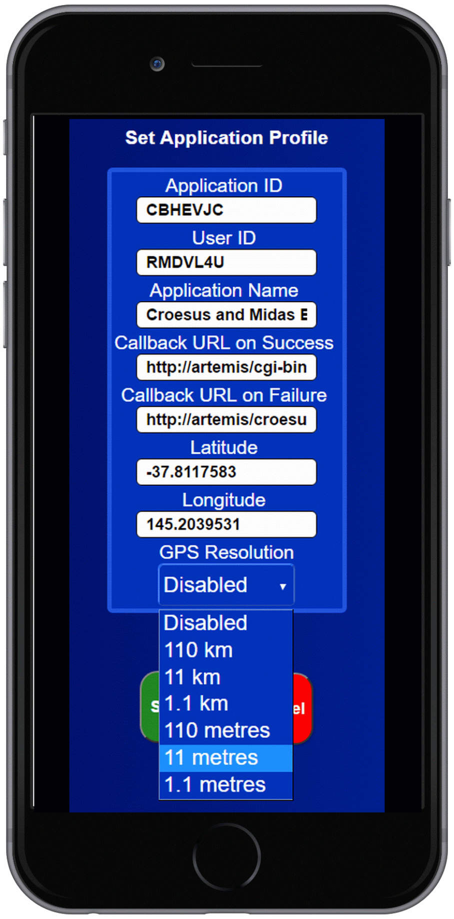
Let’s look at the overall security sequence we now have in place:

1. The user ID
2. The password
3. The device signature
4. The device location

The user ID and password may be compromised by determined attackers with too much time on their hands. Having done so, they still need the device which was registered to the service or application. Since its signature is unique, they can’t synthesise it, so they steal it. Now they need to know where to stand while using it.
It is highly unlikely that the above scenario will play out.

What It Does
The way DSS Enterprise implements this third factor, is as follows.

Each user of the system has the ability to register a latitude and longitude figure, to be associated with each service or application to which he is subscribed. Additionally, the user can specify a tolerance figure, ranging from 110 kilometres, to 1.1 metres, which are the acceptable geographic window, within which he expects to be, when accessing the service.



Set Application Profile

Application ID
CBHEVJC

User ID
RMDVL4U

Application Name
Croesus and Midas E

Callback URL on Success
http://artemis/cgi-bin

Callback URL on Failure
http://artemis/croesu

Latitude
-37.8117583

Longitude
145.2039531

GPS Resolution
Disabled

A mobile app, resident on the mobile device also associated with the service, adds the device’s current coordinates to a login request page, which it loads into the device’s browser. When the user enters his user ID and sends the page, the coordinates are sent to DSS Enterprise.

On receipt of the login request, the authentication system

- Checks the coordinates associated with the service, and compares them with those received from the device. A calculation is performed, using spherical trigonometry, to determine the difference in physical distance between the registered latitude and longitude and the coordinates received.
- If either the latitude or the longitude is removed from the registered values by a distance greater than the registered tolerance, the request is denied.
- Otherwise, a password challenge is issued.
- If the correct password is returned, the device which sent it is checked against permitted devices
- Only then, is the user granted access to the service.

How It Works

For those interested in the technicalities, the geolocation function implemented in the DSS Enterprise server, is based on the ellipsoidal earth model, defined in the WGS84 standard.
The calculation of the distance between adjacent lines of longitude one degree apart is performed using the following equation:

$$\frac{PI.a.cos(Lat)}{180.sqrt(1 - ee.sin^2(Lat)}$$

Where ‘ee’ is the square of an eccentricity factor, defined by

$$ee = (a^2 - b^2) / a^2$$

and ‘a’ and ‘b’ are the lengths of the major and minor axes of the ellipse used to model the earth in WGS84. ‘PI’ is 3.14159... and ‘Lat’ is latitude, in radians – as you’d expect. In fact, it’s easy to see that the factor PI/180 in the equation is a conversion factor for radians to degrees for a half-sphere or half-circle.

The geographical distance between adjacent lines of latitude, also one degree apart, is derived from

$$\frac{PI.a(1 - ee)}{180.sqrt((1 - ee.sin^2(Lat))^3}$$

where all variables have the same meaning as before.

The complete authentication system is described in detail in the manual
www.designsim.com.au/DSS_Enterprise_Manual.pdf