**PDetective: Dusting the Web for Fingerprints**

*'Twas the study group before Christmas, and all through the group,*

*All papers were read, by Damgard and Shoup.*

*The stockings were hung, the schemes were secured,*

*With hopes that the crypto, security ensured.*

With billions of us browsing the web everyday the web has become a major source of revenue for advertising companies. By exploiting characteristics of web browsers, companies can fingerprint users and perform more targeted advertising. Most (tech savy) people are nowadays familiar with the concept of a cookie (and not the type left out for Santa on Christmas Eve). In short, this is a file left behind on your computer when you visit a particular website and is used to recognise you the next time you re-visit the website. Recent legislation has meant that websites must make clear that a cookie will be stored on your machine (see the [EU Directive on Privacy and Electronic Communications](http://en.wikipedia.org/wiki/Directive_on_Privacy_and_Electronic_Communications#Cookies)). A cookie, however, is not the only way to fingerprint a user. Mayer in 2009 [[pdf]](http://www.stanford.edu/%7Ejmayer/papers_data/thesis09.pdf) and Eckersley in 2010 [[pdf]](https://panopticlick.eff.org/browser-uniqueness.pdf) showed that it was possible to exploit characteristics such as screen dimension and the list of installed fonts, in order to invisibly fingerprint a user (see below for more details). In total Eckersley was able to uniquely fingerprint 94.2% of half a million users. In last week's study group Luke discussed the paper "*FPDetective: dusting the Web for Fingerprints*" by Acar et al. [[pdf]](http://www.cosic.esat.kuleuven.be/publications/article-2334.pdf). This paper presents a framework which can be used to detect when websites are using these (alternative) fingerprinting techniques and presents results which show that such techniques are used in the wild.

***How does fingerprinting work?***

There are several ways that device fingerprinting could be performed, the most common being based on either JavaScript or Flash.  In the case of JavaScript there are two objects which are most commonly accessed to perform fingerprinting:

* navigator - This will contain information about the browser vendor and version, plugins supported, MIME types and some information about the OS and architecture of the machine running the browser.
* screen - Information about the resolution of the monitor, and number of colours and pixel depth.

By hashing the concatenation of the content of these two objects (together with the list of plugins and MIMEtypes), Mayer was able to uniquely identify 96% of a total of 1328 clients. Eckersley extended this in the [Panopticlick](https://panopticlick.eff.org/) project by including further contents, one of which was the list of fonts. Surprisingly the list of fonts plays a big role in the uniqueness of fingerprints. As mentioned earlier Eckersley uniquely fingerprinted 94.2% of half a million users. For readers interested to see if their own browser fingerprint is unique they can visit the [Panopticlick](https://panopticlick.eff.org/" \t "_blank) website to test it out.

***FPDetective***

Acar et al. created the FPDetective framework to discover when websites are performing fingerprinting. It consists of a web-crawler using the PhantomJS browser to collect data related to JavaScript fingerprinting and Chromium for data related to flash. Next their parser was used to extract data from the obtained log files. Following this the extracted data was stored and combined on a central server. Source code and further details are available [here](https://www.cosic.esat.kuleuven.be/fpdetective/).

Using FPDetective Acar et al. analysed the top one million website (as listed by [Alexa](http://www.alexa.com/topsites)). They found that 404 of these were running JavaScript-based fingerprinting scripts, with a total of 13 unique scripts being used. One of the scripts even went a step further and tried to remove itself after running to hide any traces. Following this Acar et al. ran the experiments to look for flash-based fingerprinting and found that 95 out of a total of 10000 websites were active in doing so. Interestingly one of these was the Turkish Ministry of Education.

***Mitigation techniques***

The [Tor](https://www.torproject.org/) Browser lists *cross-origin fingerprinting unlinkability* as a security goal, so unsurprisingly it has strong defences against fingerprinting. It does this by trying to make all browser sessions look the same.  It also limits font-based fingerprinting by bounding the number of fonts that a page can requests (Acar et al. found a vulnerability against this but it has since been fixed in recent versions of Tor).

[Firegloves](http://fingerprint.pet-portal.eu/?menu=6) was a proof of concept Firefox extension created for research purposes which tries to randomise the fingerprint. Acar et al. found that it was quite ineffective against preventing fingerprinting but since it is no longer maintained this could be because of changes to the Firefox extension system. Ironically since Firegloves itself is detectable this could actually become part of the data used to fingerprint a user due to its limited user base (1750 users).

"[Do Not Track](http://en.wikipedia.org/wiki/Do_Not_Track)" (DNT) is an HTTP header field which allows users to request that websites do not perform any tracking and is implemented in many modern web browsers. Acar et al. ran their experiments again, this time with the DNT field set to 1, but found that this yielded exactly the same results. This shows that websites are simply ignoring user's tracking preferences.