

Browser versions carry 10.5 bits of identifying information on average

[forthcoming blog post]

Technical Analysis by [Peter Eckersley](#)

This is part 3 of a series of posts on user tracking on the modern web. You can also read [part 1](#) and [part 2](#).

Whenever you visit a web page, your browser sends a "User Agent" header to the website saying what precise operating system and browser you are using. We recently ran an experiment to see to what extent this information could be used to track people (for instance, if someone deletes their browser cookies, would the User Agent, alone or in combination with some other detail, be enough to re-create their old cookie?).

Our experiment to date has shown that the browser [User Agent](#) string usually carries 5-15 bits of identifying information (about 10.5 bits on average). That means that on average, only one person in about 1,500 ($2^{10.5}$) will have the same User Agent as you. On its own, that isn't enough to recreate cookies and track people perfectly, but in combination with another detail like an IP address, geolocation to a particular ZIP code, or having an uncommon browser plugin installed, the User Agent string becomes a real privacy problem.

User Agents: An Example of Browser Characteristics Doubling As Tracking Tools

When we analyse the privacy of web users, we usually focus on user accounts, cookies, and IP addresses, because those are the usual means by which a request to a web server can be associated with other requests and/or linked back to an individual human being, computer, or local network.

Typical advice for improving your privacy as you surf the web might include blocking or deleting cookies (and [supercookies](#)), and using proxy servers or tools like tor to hide your IP address.

It's not intuitive that a User Agent poses a similar risk to a unique tracking cookie. After all, there are probably millions of people out there who use the same browser and operating system that you do. But let's examine the matter more closely. A typical User Agent string looks something like this:

```
Mozilla/5.0 (Windows; U; Windows NT 5.1; en-US; rv:1.9.1.3)
Gecko/20090824 Firefox/3.5.3 (.NET CLR 3.5.30729)
```

In fact, that was the most common user agent string among browsers visiting the EFF website during the test period: Firefox 3.5.3 running on Windows XP. There are a lot of things that can vary inside that string, and those variations can be used to distinguish and track people as they browser the Web.

Our Results to date on User Agent Identifiability

We ran an experiment to measure precisely how identifying the User Agent strings would have been among a 36-hour anonymized sample of several million requests to the EFF website. The following table shows different classes of browser, with the number of bits for best and average case User Agents within that class:

Identifying information in various classes of browsers

Browser class	Avg. identifying information	Minimum identifying information	(Least identifying user agent)
Modern Windows Desktops	10.3-11.3 bits	4.6 - 5.0 bits	Mozilla/5.0 (Windows; U; Windows NT 5.1; en-US; rv:1.9.1.3) Gecko/20090824 Firefox/3.5.3 (.NET CLR 3.5.30729)
Internet Explorer	13.2-13.5 bits	6.3 - 7.2 bits	Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1; SV1)
Firefox	8.6 - 9.4 bits	4.6 - 5.0 bits	Mozilla/5.0 (Windows; U; Windows NT 5.1; en-US; rv:1.9.1.3) Gecko/20090824 Firefox/3.5.3 (.NET CLR 3.5.30729)
Chrome	7.5-8.5 bits	5.7 - 6.2 bits	Mozilla/5.0 (Windows; U; Windows NT 5.1; en-US) AppleWebKit/532.0 (KHTML, like Gecko) Chrome/3.0.195.27 Safari/532.0
Linux	11.8-13.15 bits	6.6-7.9 bits	Mozilla/5.0 (X11; U; Linux i686; en-US; rv:1.9.0.14) Gecko/2009090216 Ubuntu/9.04 (jaunty) Firefox/3.0.14
Ubuntu	9.6 - 11.7 bits	6.6 - 7.8 bits	Mozilla/5.0 (X11; U; Linux i686; en-US; rv:1.9.0.14) Gecko/2009090216 Ubuntu/9.04 (jaunty) Firefox/3.0.14
Debian	13.5-15.3 bits	10.50 - 11.7 bits	Mozilla/5.0 (X11; U; Linux i686; en-US; rv:1.9.0.14) Gecko/2009091010 Iceweasel/3.0.6 (Debian-3.0.6-3)
Macintosh	8.8-9.3 bits	5.8-5.8 bits	Mozilla/5.0 (Macintosh; U; Intel Mac OS X 10.5; en-US; rv:1.9.1.3) Gecko/20090824 Firefox/3.5.3
iPhone	10.8 - 11.3 bits	8.7 - 9.3 bits	Mozilla/5.0 (iPhone; U; CPU iPhone OS 3_1 like Mac OS X; en-us) AppleWebKit/528.18 (KHTML, like Gecko) Version/4.0 Mobile/7C144 Safari/528.16
Blackberry	14.7 - 15.5 bits	12.0 - 12.7 bits	BlackBerry9530/4.7.0.148 Profile/MIDP-2.0 Configuration/CLDC-1.1 VendorID/105
Android	14.4 - 14.4 bits	12.2-12.4 bits	Mozilla/5.0 (Linux; U; Android 1.6; en-us; T-Mobile G1 Build/DRC83) AppleWebKit/528.5+ (KHTML, like Gecko) Version/3.1.2 Mobile Safari/525.20.1

Methodology

We took a 36 hour sample of anonymized requests to the eff.org web server by hashing the IP address of each request with a random salt, and throwing away the salt. We then calculated the amount of identifying information conveyed by each browser. Identifying information is measured

in "bits of entropy", and says how large a crowd the information would reveal you within. Browsers usually convey between 5 and 15 bits of identifying information, about 10.5 bits on average. 10 bits of identify information would allow you to be picked out of a crowd of 2^{10} , or 1024 people. 10.5 bits of information identifies can identify people from crowds of just under 1,448.

Because we did not use cookies or any other mechanism to distinguish between repeat and new visitors, each measurement of bits of identifying information lies between an upper and lower bound.¹

There are several remarkable facts about this dataset. Overall, it's amazing how identifying User Agent strings are. 10.5 bits is about one-third of the total information required to identify an Internet user.

It's also surprising that platforms like Firefox and Ubuntu, which have lower market penetration, are on average comparable or even less identifying than Windows and MSIE, which have very large userbases and should therefore have larger crowds to hide in. Part of this may be that visitors to the EFF website are over-representative of the former groups, but it's also clear that a large part of this is that Internet Explorer has a very high level of variation in its User Agent strings, with typical examples looking something like this:

```
Mozilla/4.0 (compatible; MSIE 8.0; Windows NT 6.0; Trident/4.0;
SLCC1; .NET CLR 2.0.50727; Media Center PC 5.0; .NET CLR
3.5.30729; .NET CLR 3.0.30618)
```

All of the different library and component versions there essentially function as partial tracking tokens.

Conclusions and Further Work

An average browser User Agent string contains around 10.5 bits of identifying information. This is about one-third of the entropy required to uniquely identify a human being on the planet earth.

It is not clear what low-impact steps can be taken to reduce this privacy risk. The danger with simply changing the User Agent string to something seemingly innocuous is that it needs to be done perfectly. Configuring a Blackberry to pretend to be Firefox on Windows is fine, so long as the web server can't *see* that it's a Blackberry pretending to be Firefox, since that combination will be unique or close to it.

The EFF is planning to launch a site hosting further experiments to measure the identifying information conveyed by other similarly distinctive browser features beside the User Agent string. Plugins and plugin versions are likely to be the most significant of these, and will probably reduce entropy by a similar degree to the User Agent string.

¹ One bound is based on a count in which each hashed IP address is counted for only one request; the other bound is based on treating each hit as a unique browser. In almost all cases, the true amount of identifying information pertaining to the *browser* should lie between these two values.