Firefox takes a bite out of the canvas 'super cookie'

30 OCT 2017 11

Firefox

Previous: The iOS privacy loophole that's staring y...

Next: Monday review – the hot 17 stories of the week

by Mark Stockley

Firefox 58, that's the next but one version of the browser you all trust but don't use, is going to become the first of the major browsers to do something about canvas fingerprinting – a devious, cookie-less way of tracking you on the web.

Canvas fingerprinting relies on websites being able to extract data from HTML `<canvas>` elements silently. In future Firefox users will be asked to give their permission before that extraction can take place, just as users of the Tor Browser are.

The similarity in behaviour to Tor Browser is no accident. That privacy-first browser is actually based on Firefox ESR (Extended Support Release) and a trickle of Tor Browser features and settings have been flowing slowing back upstream and into Firefox for a while now.

In the case of this simple feature, four years slowly.

So let's look at why it's better late than never.

Browser fingerprints

Browser fingerprinting has risen to prominence in recent years as the go-to approach for companies who want to track you without giving you a say in the matter.

It works by tracking your browser itself, rather than by tracking a beacon that's placed on your browser, such as a cookie, Flash LSO (local shared object) or DOM storage value.

Beacons can be blocked or deleted, fingerprints can't.

Fingerprints use information that's gathered passively from your browser such as the version number, operating system, screen resolution, language, list of browser plugins and the list of fonts you have installed.

There are many different ingredients that can be used to make up a fingerprint but the more ingredients that are included, and the more entropy available from each one, the easier it is to tell your browser from anybody else's.

One of the most popular ingredients uses the HTML `<canvas>` element.

Canvas fingerprints

The `<canvas>` element is, as you might guess, a surface a browser can draw on.

In canvas fingerprinting your browser is given instructions to render something (perhaps a combination of words and pictures) on a hidden canvas element. The resulting image is extracted from the canvas and passed through a hashing function, producing an ID.

Different graphics cards and operating systems work slightly differently, which means that if you give two different website visitors identical drawing instructions, they'll actually draw slightly different pictures.

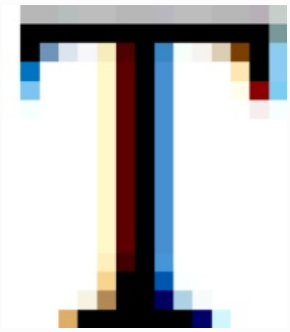
Complex instructions can produce enough variation between visitors to make canvas fingerprinting a potent ingredient in a fingerprinting recipe.

The more complex the instructions, the easier it is to tease out differences between individuals' browsers, but the basic principle can be seen with a simple test.

The pictures below show the letter T as rendered by Firefox (left) and Safari (right) on my

system, with hashes of the images shown beneath. The differences are just about visible, but all that really matters for the purpose of fingerprinting is that they aren't exactly the same and will therefore produce different hash values.

55b2257ad0f20ecbf927fb66a15c61981f7ed8fc



17bc79f8111e345f572a4f87d6cd780b445625d3

A step in the right direction

Fingerprinting is difficult to stop because it turns the complexity, customisability and openness of modern browsers against them. The more personalised your browser is, and the more willing it is to share information about itself, the more it stands out in a crowd.

Plugins can help by intercepting known fingerprinting scripts, but they also make things worse by adding entropy to your browser's fingerprint.

A balance needs to be struck between the usefulness of any given feature and its potential for abuse. Browser vendors also need to stay on top of how features are actually being used, rather than how they're supposed to be used.

A case in point is the Battery Status API. The feature exists so that "web developers are able to craft web content and applications which are power-efficient". In fact the ability to determine which of 14,172,310 different levels of charge your battery is at has been largely ignored by developers, but adopted enthusiastically as a fingerprinting technique.

About a year ago it was summarily dumped by Firefox.

To combat canvas fingerprinting the Firefox developers have opted for the pragmatic opt-in approach of Tor Browser instead of outright rejection. That's because although canvas fingerprinting is a bigger problem than battery status abuse, dropping `<canvas>` isn't an option. It isn't a white elephant like the Battery Status API is, it's actually a fantastically useful feature, but just happens to be a very popular fingerprinting technique too.

At least for a few more months, anyway.

browser fingerprint | Canvas Fingerprinting

Previous: The iOS privacy loophole that's staring y…

Next: Monday review – the hot 17 stories of the week

About the author

Mark Stockley ▸

11 comments on "Firefox takes a bite out of the canva…"

BillBlagger October 30, 2017 at 11:04 am

Thanks. Interesting and understandable.

6 0 Rate This

Reply

---

jules October 30, 2017 at 12:33 pm

Will Canvas Fingerprinting be permissible under GDPR in the EU?

0 0 Rate This

Reply

Paul Ducklin October 30, 2017 at 1:11 pm

IANAL, but I guess the answer is “it depends”.

If all you’re doing is figuring out something about the browser to help your own JavaScript make good decisions about how to format and render content for greater legibility, and you don’t keep the “fingerprint” after the user leaves your site, why not? GDPR won’t (as far as I know) ban you using tell-tale information such as Referer headers or other HTTP data that reveals your broswer type, operating system, or whatnot.

0 0 Rate This

Reply

James (@jholyhead) October 30, 2017 at 4:34 pm

If the information is in any way personally identifiable (and it easily could be, depending on what other information you are gathering), then yes, absolutely, it becomes personal data and is subject to GDPR regulations.

0 0 Rate This

Reply

Mark October 30, 2017 at 12:53 pm

At least 5 years too late…

0 1 Rate This

Reply

Kenneth October 30, 2017 at 1:35 pm

Sometimes when doing tasks, it jumped to the survey’s ‘special debug screen’, showing all my browser information, detailed. I frowned.

0 0 Rate This

Reply

Emanuel Mosi October 30, 2017 at 2:07 pm

Nice

0 0 Rate This

Reply

Martin Kopser October 30, 2017 at 3:18 pm

Do you mean the differences are just about *in*visible?

“The differences are just about visible, …”

0 0 Rate This

Reply

Mark Stockley October 30, 2017 at 3:27 pm

No, I meant they are just about visible. I can see them, but it took a bit of effort.

2 0 Rate This

Reply

Paul Ducklin October 30, 2017 at 4:52 pm

Those screenshots are seriously zoomed in – at normal pixel pitch the differences are as good as invisible. (Each of the big coloured squares in each of the images corresponds to one dot on your display. The differences seem to be down to pixel-level minutiae such as how subpixel antialiasing is configured, what border colours are nearby, and which font rendering library code was used by the browser vendor.)

0 0 Rate This

Reply

Alex October 30, 2017 at 4:29 pm

“Firefox 58, that’s the next but one version of the browser you all trust but don’t use”

Conway’s Law at work. Mozilla as an organization has always been focused on such attributes as “privacy” and “security”, while de-prioritizing issues as “usability” and “consistency”.

They’ll spend years working on a great privacy feature like this (bravo, truly!), while ignoring the fact that keyboard shortcuts have been broken for 10 years. It should come as no surprise that Firefox is admired but mostly ignored.

I love what Mozilla is doing with security and performance these days, but until they can make their browser less painful to use, I can’t switch back to it for daily use.

0 0 Rate This

Reply

Leave a Reply

Enter your comment here...

Recommended reads