SUBJECT: Please do enable DoH in the UK
FROM: Richard Neill <[redacted]
TO: [redacted]
DATE: 07/12/2020 08:51

Dear Mozilla,

I have always been a great fan of Firefox, but not of our current UK government which takes a very hostile approach to privacy. The fact that UK Government is against DoH should be a powerful reason in favour of rolling it out, protecting user privacy of UK citizens.
Furthermore, anything that can give users back their privacy, even on corporate networks, is a good thing: users have coerced-consent (they have to agree in order to work there).

In terms of this, you should know that GCHQ are "the bad guys". We know, from Snowden, that governments do not always wield their power responsibly.
https://www.zdnet.com/article/mozilla-no-plans-to-enable-dns-over-https-by-default-in-the-uk/
Furthermore, while there is a good case to block child abuse images, I think copyright infringement is far less important than privacy and free speech, and while extremist material may be problematic, driving it underground (it's still accessible via Tor etc) can also make the problem worse - because it makes it harder for non-extremists to see, and therefore counter this material.

Best wishes,

Richard Neill

--
Chief Digital Officer,
Unipart Digital

P.S. As an aside, while you are working on privacy, can you please also consider a way for the server to detect the SSL fingerprint that the browser sees. i.e. we need a way that a conscientious webserver administrator can protect the privacy of his users against a malicious HTTPS-interception/inspection device. For example.

* Alice runs a webserver (and wants to take responsibility for users' privacy)

* Bob is a naive user, running Firefox, on a corporate-provided machine which has a fake SSL-trust certificate installed, originating from the SSL snooping appliance installed by Eve.

* Eve is the company sysadmin. She believes she is trying to protect Bob from himself, but is harming Alice who does not consent to the interception on her behalf, nor the communications of the other users who communicate via her server with Bob.

Some possibilities might be:
1. Return the SSL key fingerprint it thinks belongs to Alice (and which is actually Eve's) with the HTTPS headers (in some signed way so that it can't be tampered with).
2. Make that SSL key fingerprint available to Javascript (or have a JS function that can at test if a particular fingerprint is the the real one)
3. Have an external way to verify the SSL key fingerprint of a public server.

As a result:
* Ideally, Bob gets proper encrypted communications.
* Or Alice has a way of finding out that her messages are being intercepted (and can warn Bob "Don't connect from this machine, it's compromised").
* Or at the very least, Bob should get a warning showing that his SSL chain of trust has been compromised, similar to the "Broken padlock" icon.