Hi,

I'm responding to the Mozilla question about DNS-over-HTTPS implementation.

Respecting privacy and security

1. Data retention duration
   My understanding is that, operationally, a DNS operator only needs to retain
   the association between a client IP/port and a DNS query while responding to
   that request. However, DNS responses should be cached as specified by the
   originating server.

   Thus, user data should not be retained longer than the time needed to answer
   the DNS query.

2. Data collection in emergency circumstances
   Since the resolver should not be able to know who the user is, I have a hard
   time seeing a situation where it would be useful to collect more data.

3. Third-party audits
   No special comment here

4. Transparency report
   I feel that the format of the transparency report should be explicitly stated,
   and designed in an easy-to-parse format.

Online Safety

1. Domain filtering
   Domain filtering might be mandated by law; the operator has to comply.
   Mozilla could set up a program to decide that a given operator is filtering
   too many domains to be allowed to continue operations, in which case the rules
   should be clearly stated in advance.

   Moreover, a blocked domain should return a clear reply to the end user:
   "This request has been blocked by government decision ...". An operator
   should only
   block domain explicitly blocked by administrative orders, not based on
Regarding cross-border filtering, I would suggest that, if an operator wishes to provide DNS resolution service to residents from another country, they set a specific endpoint up for said country, where only the filtering rules for said country would be applied — basically, use `doh-uk.example.org` for users from the UK, and `doh-us.example.org` if the US rules have to be applied.

2. Harmful outcomes from blocking at the DNS
Blocking at the DNS level is dangerous: administrative orders might force blocking unrelated sets of hosts or services (e.g. "Block github.com because of this specific repository"), and won't prevent motivated end users from accessing the content — they would rely on another resolver, or retrieve a custom `hosts` file from another source.

3. Effective means of protecting users
Since we're talking only about the browser, this could be handled by the browser, which could fetch a list of dangerous websites from an authoritative source and add warnings (or fetch a custom certificate revocation list).

4. Transparency and accountability
Require operator to publish, in a standard format, the list of currently blocked / filtered domains.

5. Opt-in filtering
When the user decides to enable DOH (Which should be a voluntarily action), provide a list of potential providers, with a set of options for each of them. Include links to the related policies. If DOH is in use, add a simple to use notification (maybe close to the shield in the URL bar) describing the current DOH settings, provider, and category.

Building a better ecosystem

1. Trust in Internet Technologies
DoH reduces trust in Internet Technologies: it pushes a narrative of...
"Your ISP is evil"; instead of trying to push said ISP to fix its act, it bypasses it altogether. Moreover, since that mechanism is specific to an application, it could lead to some sites or services working differently depending on the application, in a hard to read way — users might have a worse experience using some services through Firefox since DoH might break geoIP DNS optimizations, leading to them loading content from far away CDN nodes instead of those at their ISP.

Moreover, it is bound to break some local LAN services, bringing a feeling of "DoH is breaking my home" or "Don't use Firefox, it's broken".

2. DNS exploitation in my region
I am not aware of any DNS exploitation risks in France.

3. Gaining DoH adoption among ISPs and DNS providers
- Define a way for DoH to work smoothly with GeoIP DNS lookups for CDN/cache optimization;
- Define a way for DoH to work properly with local LAN DNS suffixes (might not always be `.local`)
- Work to have DoH handled on the local gateway instead of per-application
- Work to have DoH handled OS-wide, not per-application

4. DNS use cases where DoH provides security and privacy
Encrypting DNS requests is only a mild protection against a network-level attacker: they still see all packets and their target, and can spy on SNI TLS negotiation.

The current design of DoH (at the application level) seems broken to me: it doesn't allow for network operators to configure DNS on their networks to provide local services. In order to address that need, a special DNS entry can be set to disable it locally (use-application-dns.net); any network operator wishing to spy on its users can simply set said entry on their LAN to disable DoH.

If the goal is to provide security, DNSSEC validation would be sufficient. For privacy, DNS-over-TLS provides said privacy between a device and
its chosen resolver; a user is, however, better served with a VPN to their remote host of choice.

5. Issues with DoH deployment
When a local network uses split-horizon or local DNS entries (e.g. for a printer, or some file servers), while not fully managing each employee's device, DNS resolution to said printer will break.

This also breaks when a user is using a partial DNS, which uses per-connection DNS servers and searches: the OS is able to route DNS request for a specific domain to the right DNS server based on said configuration; on the other hand, an application doesn't have this knowledge and would send all queries (including those for a VPN-only domain) to remote servers. Specifying a DoH server over the VPN is still an issue, since it would send all local queries to that VPN-based DoH server, whereas a system-level setup would only send queries for the relevant domains to the VPN-based DNS server.

It also prevents DNS64 from working, which would be required when the local network is IPv6-Only.

It makes it harder for users to provide a single filtering DNS for their home (e.g. pihole), since they would have to disable or reconfigure DoH for each device instead of a single, global configuration in their DHCP / RA daemon.

Bypassing the ISP DNS stack prevents users from benefitting from GeoIP cache finding, which decreases the end user experience significantly - unless the service they are accessing is provided by a provider who has been able to set up peering/BGP-based cache routing.

Operating at an application level instead of a system-wide setting causes inconsistency between applications: a website might work on Firefox but not in an Electron app, etc.

Summary
While DoH is an interesting concept, and could be an interesting alternative to DoT + DNSSEC, it should not live at the application level. It should be moved to a system-wide service, which users can configure or control on their own, while being able to interact with network-level DNS configuration.

The ideal model would be:
- I can decide to use a specific DoH server or the network-based DNS host for each "main" connection (home, office, 4G, other);
- When I enable a VPN, the system receives the list of domains to send to that VPN connection's DNS server — and I can decide if I agree or not;
- I have a single control panel where I can see important details about my DNS setup: details about available DoH servers, their configuration, etc.
- By default, requests for my LAN-pushed search / domain suffixes go to my LAN DNS server;
- This application is used as a system-wide DNS resolver;
- I can add block lists manually to the application;

If you need some details about some of my answer, please feel free to contact me.

Regards,

--
Raphaël Barrois