DNS Privacy Protocols

Encrypted DNS queries for privacy protection

Klaus Möller

WP8-T1

Webinar, 10th of December 2020

Public

www.geant.org
What we will cover today

• DNS over TLS (DoT)
• DNS over HTTPS (DoH)
• DNS over QUIC (DoQ)
  - Resolverless DNS
• Considerations & Recommendations
DNS Threats

• Haven’t been entirely correct last time
  – Most stub resolvers don’t do full DNSSEC validation by themselves
  – That means, stub-resolvers have to trust the cache/forwarder
  – And the last leg between stub and cache will have to be secured

• Enter DoT & DoH
  – And in the future: DoQ
Why protect the last leg?

- Ability to interfere with DNS lookups is/was widely abused
- ISPs redirecting to domain selling sites, etc.
- Parental controls, i.e. blocking adult content, etc.
- Governments
  - Regime criticism in authoritarian regimes: North Korea, China, Russia, etc.
  - Various reasons in Western Democracies (i.e. UK, Germany, etc.)
    - Parental controls, child pornography (UK, Germany)
    - Hate speech, Nazis, Islamic State propaganda (Germany)
    - Black markets, Wikileaks, and more on the wish list ...
- Power users, home network: Ad-blocking
- Network admins: Malicious Site blocking
DNS over TLS
DNS over TLS (DoT)

• Use DNS over either
  – UDP with DTLS (MAY support, RFC 8094)
  – TCP with TLS (MUST support, RFC 7854)

• Port number in both cases: 853

• TLS version used will be the most recent one, currently 1.3

• Protocol is otherwise the same

• Scope
  – Stub Resolver to Caching Resolver
  – Zone Transfer
  – Dynamic Updates
DoT Usage Profiles (for DTLS/TLS)

- **Strict Privacy profile (RFC 8310)**
  - Requires an encrypted connection and successful authentication of the DNS server
  - Mitigates both passive eavesdropping and client redirection
  - But no DNS service if an encrypted, authenticated connection is not available

- **Opportunistic Privacy profile (RFC 8310, 7858)**
  - Attempts, but does not require, encryption and successful authentication
  - Limited or no mitigation for above attacks but maximizes the chance of DNS service
  - Initial queries (for IP address of the DoT server) use this profile
DoT: Trust the server key problem

- Trust the certificate chain from the CAs or not
- What if your certificate store is poisoned with a Man-in-the-Middle certificate?
  - So that firewalls/IDS/IPS can break up TLS traffic
  - But will you still have web access without that certificate?
- Names in the certificate (chain) require opportunistic lookup
  - Unless Auth name is learned out of band
DoT Client Support

- **Linux**
  - Not covered by glibc (and will likely never be)
    - nss-tls supports only DoH, plugs-in through Name Service Switch (NSS)
  - Locally run resolver daemons:
    - systemd-resolved, NLnet Labs stubby daemon (getdns), Knot Resolver, ...

- **Windows**
  - Not covered directly (support announced, but DoH will come first)
  - NLnet Labs stubby daemon

- **iOS 14**
  - No user configuration of servers without 3rd party tools

- **Android 9 (Pie)** – off by default
  - Apps mostly add somewhat more comfortable UI to change the server
DoT Server Support

• Nameservers
  – PowerDNS Dnsdist (1.3.0)
  – Unbound (01/2018)
  – Knot Server
  – Etc.

• Nameservers without support (yet)
  – Windows DNS server
  – BIND
    • Stunnel as workaround
    • Proposals for BIND 9.17, but no code as of now
DNS over HTTPS

www.geant.org
DNS over HTTPS

- Use Cases (as per RFC)
  - Preventing on-path devices from interfering with DNS operations
  - Allowing web applications to access DNS information via existing browser APIs in a safe way consistent with Cross Origin Resource Sharing (CORS)

- More limited than DoT, only the path between (stub) Resolver and RDNS/Cache
DoH Technical

- DNS operations accessed via URL template
- Examples:
  - https://dns.google.com/dns-query (POST)
  - https://dns.google.com/resolve? (JSON)
- Configuration, discovery, and updating of the template not part of the protocol
- Only redirect code 301 (moved permanently) currently supported
- HTTP/2 allowed
  - Recommended for performance: reordering, parallelism, priority, header compression
  - Server Push may be used to send answers in advance to client
DoH Query Methods

- **GET**
  - `dns-query?dns=BASE64URL_OF_QUERY`
  - Base64URL schema is different from plain Base64 (see RFC 4648, sec. 4)

- **POST**
  - Query will be transmitted as Base64 encoded DNS message
  - Content Type: `application/dns-message`
  - Should be used with care, as return data may not be cached

- **JSON**
  - All queries use GET method
  - DNS query parameters: `name`, `type`, `cd`, `do`, `edns_client_subnet`, `random_padding`
  - Response can be JSON (`application/x-javascript`) or binary (`application/dns-message`) determined by `ct` parameter
DoH & Proxies

- HTTP proxies & caches are allowed and supported by DoH
  - Of course, MitM SSL proxies can see all queries
- Oblivious DoH (proposal from Cloudflare)
  - HTTPs between Client – Proxy and Proxy – DoH server
  - Additional query encryption between client and DoH server
- But it does very little with regards to privacy
  - DoH server will know question & answer, source IP address is incidental
- Lots of ways to leak client addresses due to implementation errors
  - EDNS subnet options (client)
  - DNS XDF pseudo RR (client)
  - X-Forwarded-For HTTP Header (proxy)
- How to be sure that proxy and server do not collude?
DoH Problems

- Correlation through
  - Long lived TCP connections
  - TLS session resumption
  - HTTP headers (Auth, User-Agent, Accept-Language)
- Traffic analysis about queries possible if no/false padding or no compression is used
- EDNS client subnet option should not be used in queries
- No OCSP, AIA lookups or deadlocks may happen
- Chicken or the egg problem for name of DoH server
DoH in Browsers: Chrome

- Chrome “Secure DNS”, starting with Chrome 83
  - `chrome://flags#dns-over-https`
  - Seems to be unavailable on Linux

- Available on Android and Windows and enabled
  - Default: Use system DNS server, try to use it with DoH
  - Silent fallback to normal DNS lookups in case of problems

- Policies available for managed environments
  - `DnsOverHttpsMode`, `DnsOverHttpsTemplates`
DoH in Browsers: Chrome-based

- Similar procedure for Edge, Brave, Opera, etc.
  - Substitute `chrome://` with `edge://`, `brave://`, etc.
DoH in Browsers: Firefox

- “Trusted Recursive Resolver (TRR)”
  - Opt-out, not opt-in!

4. The user will be informed that we have enabled use of a TRR and have the opportunity to turn it off at that time, but will not be required to opt-in to get DoH with a TRR.
Firefox TRR settings

- Complex heuristic
  - Look for `use-application-dns.net. domain`
  - Look for enterprise or user settings
  - `security.enterprise_roots.enabled` allows installing private root certificates
    - For breaking up of HTTPS by content filtering proxies,
    - I.e. your lookups aren’t secret anymore then

- Fine grained control
  - `about:config`
  - `network.trr.*`
Other DoH Implementations

• Supported Client OS
  − Android 9 (Pie)
  − Apple iOS 14
  − Apple macOSX 11

• Not yet supported Client OS
  − Linux glibc (and will likely never be), see DoT
  − Windows: announced
    • Insider Preview Build 19628
    • Configuration GUI with Insider Preview Build 20185

• Nameserver
  − Unbound, Knot DNS, CoreDNS, Technetium DnsServer, ...
DNS over QUIC

Resolverless DNS

www.geant.org
QUIC – Quick UDP Internet Connections

- Many transactions have a simple Request – Response pattern
  - But setting up a TCP connection with TLS on top requires several round trips before data can be sent

- Need for a protocol that has fewer round-trips: QUIC
  - Combine TLS and TCP handshake in one setup
  - And take the flow control from TCP up to the application
  - Will use UDP, port numbers may be different from existing applications

- Meant as a replacement/supplement for TCP + TLS or UDP + DTLS
  - Invented by Google, now an IETF standard
  - Standardization not finished, incompatible implementations as yet
  - HTTP/3 will be defined on top of QUIC
QUIC – Round trip savings

Zero RTT Connection Establishment

TCP

100 ms

TCP + TLS

200 ms¹
300 ms²

QUIC
(equivalent to TCP + TLS)

0 ms²
100 ms²

1. Repeat connection
2. Never talked to server before

Source: https://blog.chromium.org/2015/04/a-quic-update-on-googles-experimental.html
DNS over QUIC

- Same principle as with DoT or DoH
  - Internet draft as of now
  - Port number not yet decided, maybe 784/udp?

<table>
<thead>
<tr>
<th></th>
<th>UDP</th>
<th>TCP</th>
<th>TLS</th>
<th>DTLS</th>
<th>QUIC</th>
</tr>
</thead>
<tbody>
<tr>
<td>Transport efficiency</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Connection set up time</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>0-RTT</td>
</tr>
<tr>
<td>Head of queue blocking</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Retransmission efficiency</td>
<td>x</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Long messages (DNSSEC)</td>
<td>x</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Security</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Three ways handshake</td>
<td>x</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Encryption &amp; Authentication</td>
<td>x</td>
<td>x</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
</tbody>
</table>
Resolverless DNS?

• Idea: DNS Responses are pushed from web servers to the clients
  - Through the HTTP connection
  - No DNSSEC, TLS considered safe enough

• No resolver needed, henceforth “resolverless DNS”

• Motivation/Use Case:
  - Web content includes lots of references to other objects (Pictures, Videos, Ads, etc.)
  - DNS lookups for their sites takes round-trips and thus time
  - And allows Ad-blocking

• Bad idea, because
  - Ties DNS to Web content providers, esp. the very big ones, even more
  - Web site defacement will now mean DNS cache poisoning too
  - Circumvents BHDNS protections and Ad blocking
Considerations & Recommendations

www.geant.org
Technical considerations

- Problems with using external RDNS
  - Answers will have the external view of the network not the internal
  - All other programs, even those started by browsers, still use the system resolver
  - Thus, results returned may/will differ → hard to debug problems
  - Additional work for opting out: configuration, canary domains, ...

- External RDNS cannot know why (local) DNS manipulation is done
  - Parents
  - Sysadmins
  - Security teams (PDNS monitoring)
  - Governments
  - ISPs
Privacy considerations

- None of them (DoT, DoH) really protects privacy
  - Still can see the metadata of your connections (even with HTTPS)
    - Eavesdropper can infer from metadata what’s been queried in DNS
  - Queries coming from the recursive resolver are not encrypted

- Won’t help against evil governments
  - Need a VPN (and more) for that
  - If you have a VPN (a trustworthy one), what value do “DNS over ...” add?
  - They have the resources to block DoH (they already block a lot more)
Privacy considerations (cont.)

• Why trusting your local DNS servers is better
  – Big tech companies track record w/ regards to privacy
  – Big (central) data pools will raise desires from governments
    • Tech C. usually budge after some phony resistance
  – Much more leverage against your admins/employers
    • Same jurisdiction
    • Better legal situation (employee rights, GDPR, etc.)
    • At least in Western Europe

• Different situation as ISPs
  – Neutrality obligations?
  – Business opportunities
Political considerations

- DoH & resolverless DNS are political solutions
  - Add nothing security-wise (compared to DoT/DoQ)
  - Add nothing privacy-wise (compared to DoT/DoQ)
  - But breaks Split DNS, RPZ, PDNS, ...
  - No added value (for end-users, network admins)
  - Web servers will force their view of the network upon end-users
  - Power of big tech companies will grow even more

- Network landscape
  - Endpoints are insecure and will be so in the future
  - Need to allow/block some kinds of traffic – through Firewalls, DNS, web-proxies

Source: https://blog.powerdns.com/2019/12/03/doh-anti-competitive-and-network-neutrality-aspects/
Recommendations

● For managed networks
  – Block outbound DNS (ports 53, 853, UDP & TCP)
  – Block outbound DoQ (whatever port it will be)
  – Block IP addresses of known DoH providers
  – 1.1.1.1, 4.4.4.4, 8.8.8.8, 9.9.9.9, ..., list is short enough (i.e. Cisco Umbrella)
    • Might discourage unreasonable users/vendors
  – Or force all HTTPS traffic through MitM proxy :((

● Use DoT/DoQ (even DoH) with internal RDNS
  – Can still use PDNS, RPZ, Split DNS

● May use DoH servers at home/on your device
  – If so, check for servers DNSSEC support, logging & filtering
What have you learned?

The Good
• DoT
• DoQ

The Bad
• DoH
• Resolverless DNS

Things that have been left out
• DNSCurve
• DNSCrypt
• DNS Protocol Details (EDNS)
• Response Rate Limiting (RRL) → part of upcoming DDoS course
Thank you

Any questions?

Next course: Distributed Denial of Service Protection
8\textsuperscript{th} of February 2021

www.geant.org
References:

- PowerDNS blog: “DoH: (Anti-)Competitive and Network Neutrality aspects”
  https://blog.powerdns.com/2019/12/03/doh-anti-competitive-and-network-neutrality-aspects/
- National Cyber Security Center: “Factsheet DNS Monitoring will get harder”:
- Zdnet: “DNS-over-HTTPS causes more problems than it solves, experts say”,
- Elbsides 2019 session featuring vixie (pro DoT), Michaelis (pro DoH) and a panel discussion afterwards:
  https://www.youtube.com/channel/UC1kRI13BZ6KMCwtGttD5Arg/videos
- Running a DNS Privacy server:
  https://dnsprivacy.org/wiki/display/DP/Running+a+DNS+Privacy+server
- Cloudflares Secure Browse Check: https://www.cloudflare.com/ssl/encrypted-sni/
- Wei & Heidemann, Whac-A-Mole: Six Years of DNS Spoofing,
Tools & Browsers

- JSON API for DNS over HTTPS (DoH) https://developers.google.com/speed/public-dns/docs/doh/json
- DoH in Firefox:
  - https://wiki.mozilla.org/Trusted_Recursive_Resolver
- DoH in Chrom* (Edge, Opera, etc.)
  - https://www.tenforums.com/tutorials/145372-how-enable-disable-dns-over-https-doh-google-chrome.html
- Public DNS Server List
  - https://github.com/curl/curl/wiki/DNS-over-HTTPS#publicly-available-servers
  - https://dnscrypt.info/public-servers/
  - https://beebom.com/best-dns-servers/
  - https://www.allconnect.com/blog/best-free-dns-servers
- Linux
  - NSS-TLS: https://github.com/dimkr/nss-tls
- List of DoT and DoH implementations:
  - https://doh.defaultroutes.de/implementations.html
RFCs