DNS-over-TLS implementations

Implementation Status

This table lists the best understanding of the current status of DNS-over-TLS related features in the latest stable releases of a selection of open source DNS software.

If there are errors or glaring omission please email sara@sinodun.com

> **Also see** guides on how to use NGINX and other proxies to provide DNS-over-TLS, also see here.

This works with a couple of provisos:

- Be aware that a client will think it is talking to a DNS-over-TLS server and so may keep connections open when idle even when not using EDNS0 Keepalive (as allowed by RFC7858). The nameserver will see only TCP connections which were historically used just for one-shot TCP and may not be robust to many long-lived connections.
- Therefore this **will work much better** if the nameserver has robust TCP capabilities (as described in Sections 6.2.2 and 10 of RFC7766), and would be required for production level service. Any server that fully implements EDNS0 Keepalive (RFC7828) should meet this criteria.

See the DNS Privacy reference material page for more details on the individual features.

Clients

Mode		Stub						Recursive resolver			
Software		ldns (drill)	digit	getdns (Stubby)	BIND (dig)	Go DNS	Knot (kdig)	getdns[a]	Unbound	BIND	Knot Res
TCP/TLS Features	TCP fast open[b]							P			
	Connection reuse (Q/R, Q/R, Q/R)										
	Pipelining of queries (Q,Q,Q,R,R,R)	n/a									
	Process OOOR (Q1,Q2,R2,R1)	n/a									
	EDNS0 Keepalive[c]										
TLS Features	TLS encryption (Port 853)										
	TLS authentication										
	EDNS0 Padding										

Servers

Mode		Recursive			Auth		
Software		Unbound	BIND	Knot Res	NSD	BIND	Knot Auth
TCP/TLS Features	TCP fast open**						
	Process Pipelined queries						
	Provide OOOR	WIP			n/a	n/a	n/a

	EDNS0 Keepalive* **	WIP					
TLS Features	**TLS encryption (Port 853)**		(d)				
	Provide TLS auth credentials		(d)				
	TLS DNSSEC Chain Extension						
	EDNS0 Padding (basic)						

KEY:

- Green square - indicates latest release already supports this functionality
- Blue square - indicates that a patch is available in our git repo. See here for details: DNS-over-TLS patches
- Yellow square - indicates work in progress, or availabe in next release
- P - Requires building against a patched version of libunbound

(a) getdns uses libunbound in recursive mode
(b) not yet available on Windows
(c) Implies robust TCP connection management (see RFC7828 and RFC7766)
(d) See this article for how to use stunnel with BIND to provide DNS-over-TLS - thanks Francis Dupont!

Note pipelining and OOOP are not applicable for synchronous applications.