RIPE NCC
RIPE NETWORK COORDINATION CENTRE

Deploying DNS over TLS for the RIPE Meeting

Colin Petrie | 16 May 2018 | RIPE 76

Important Service Announcement :)

In case you can’t guess from the topic of this presentation…..

There are DNS-over-TLS resolvers available at RIPE 76

So you should use them!
(or at least, play with them)

Colin Petrie | RIPE 76 | 16 May 2018

Why are we doing this?

- The DNS WG chairs asked us to add them
- We are nice, and like to keep WG chairs happy
- Encourage everyone to gain experience
- Also, it sounded cool and fun
  - normally, resolvers are boring
  - no-one pays attention to them except when they don’t work!

What are we using?

- Knot Resolver v2.3.0
- 4 resolvers in a load balancer pool
  - same servers as our existing Bind 9.12 resolvers
- Listening on port 853
- Same IP addresses as existing resolvers
  - 2001:67c:64:53::53:1
  - 2001:67c:64:53::53:2
  - 193.0.31.237
  - 193.0.31.238

Choosing software

- We want all the latest shiny privacy features
- We also have some operational requirements
- Wish-list:
  - DNS-over-TLS
  - Qname Minimisation
  - Aggressive use of DNSSEC-Validated cache
- Mandatory (currently supported)
  - DNSSEC validation
  - DNS64 (for our NAT64 network)

Choosing software

Servers

Mode		Load Balancer	Recursive				
Software		dnsdist[d]	Unbound	BIND	Knot Res	CoreDNS[f]	Tenta[f]
General	QNAME minimisation	n/a	✓		✓		
TCP/TLS Features	TCP fast open[b]	✓	✓	✓	✓		
	Process Pipelined queries	✓	✓	✓	✓		
	Provide OOOR	(h)		✓	✓		
	EDNS0 Keepalive[c]			✓			
TLS Features	TLS encryption (Port 853)	✓	✓	(e)	✓	✓	✓
	Provide TLS auth credentials	✓	✓	(e)	✓	✓	✓
	EDNS0 Padding (basic)			✓	✓		
	TLS DNSSEC Chain Extension						

https://dnsprivacy.org/wiki/display/DP/DNS+Privacy+Implementation+Status#DNSPrivacyImplementationStatus-Servers

- Very useful, if you know what features you require

Evaluating Bind

- Because our existing resolvers use Bind 9.12
  - No Qname Minimisation (yet!)
  - No TLS support
    - Workaround - running TLS proxy (nginx or stunnel)
      - Conflicts with ACL support for DNS64 as queries come from the proxy source address
      - To support DNS ACLs, would need to run two different proxies, or use address re-writing or source routing
  - We would make the system even more complex, fragile, and difficult to debug

Evaluating Unbound

- One daemon to resolve our entire wish-list
- Especially, terminates TLS directly - is aware of client source address (for DNS64 ACL)
- But DNS64 is a global flag - no ACL support
- We could run two different daemons, and again look at re-writing or mapping destination addresses/ports at the load balancer, etc
- But now we are back to the same problem - adding much more complexity

Evaluating Knot Resolver

- One daemon to resolve our entire wish-list
- Especially, terminates TLS directly - is aware of client source address (for DNS64 ACL)
- But DNS64 is a global flag - no ACL support (sound familiar?!)
- But DNS64 is implemented as a Lua module
- 10 lines of Lua code, to add selective source address matching to the DNS64 module
- Profit!

Testing

- Set up the load balancer pools for TCP port 853, new Knot Resolver instances added
- Manual testing using kdig
- Use as upstream recursive resolver using Stubby, tested in daily use (by me!)

- Now it is up to you!
- Details at: https://ripe76.ripe.net/on-site/technical-information/dns-over-tls-resolvers/

Qname Minimisation weirdness

- During initial testing, I noticed an issue
- For an example of ‘name.co.uk'
  - Follows referral from ‘.’ to ‘uk’
  - Queries ‘.uk’ for ‘.co.uk'
  - Gets authoritative answer
  - Shuts down minimisation
- Only minimises if the answer is a referral
- Stops minimising on authoritative answer

Qname Minimisation

- Many ccTLDs use Second Level Domains (SLDs)
- Often the TLD and the SLD are operated by the same authoritative servers
- Examples:
  - .uk (.co.uk)
  - .ke (.co.ke)
  - .nz (.co.nz)
- Shuts down minimisation for some ccTLDs :(

Qname Minimisation

- RFC 7816
- Appendix A. An Algorithm to Perform QNAME Minimisation
- (6) Query for CHILD IN NS using ANCESTOR's name servers. The response can be:
  - (6b) An authoritative answer. Cache the NS RRset from the answer section, and go back to step 1
- Step (6b) is skipped by Knot Resolver

Qname Minimisation

- Fixed by adding support for Step (6b)
- 15 line patch
- Submitted upstream:
  - https://gitlab.labs.nic.cz/knot/knot-resolver/issues/339
- This patch is on Knot Resolver instances here at RIPE 76
- Feedback! Testing! Does it work? Does it cause any problems?
- Tell CZ.NIC :) Especially if it works OK!

Recent News

- A recent development:
  - https://android-developers.googleblog.com/2018/04/dns-over-tls-support-in-android-p.html
- Latest developer preview of Android supports “Private DNS” mode
- “By default, devices automatically upgrade to DNS over TLS if a network's DNS server supports it”
- Does anyone have a device like this here? Can we check if this works with our servers?

Questions

cpetrie@ripe.net