Update on DNS Privacy Measurements

Funded by a grant from the Open Technology Fund

Sara Dickinson          sara@sinodun.com (Presenter)
John Dickinson          jad@sinodun.com
Jim Hague               jim@sinodun.com

sinodun.com             @SinodunCom
Previous work on TCP/TLS measurements

• Our work:
  • Comparisons of 4 nameservers for small number of clients (10s)
  • Varying queries per connection (including low numbers)
  • *dnspert* with UDP/TCP/TLS on 2 bare metal machines

• Baptiste Jonglez:
  • *Unbound* only but thousands/millions of cloud VM clients (6.5M!)
  • Just UDP/TCP using simple *dnsscaler* tool, connections not closed
Key Results (Sinodun)

- Using 8 clients
- Solid line is TCP, dotted is TLS
Key Results (Sinodun)

- Using 8 clients
- Solid line is TCP, doted is TLS

With 100-2000 queries per connection, queries handshake amortised.
Key Results (Jonglez)

- 30k con per client VM
- Unbound

UDP/TCP comparison
Key Results (Jonglez)

- 30k con per client VM
- Unbound

TCP is 25% of UPD with 25,000 clients
Key Results (Jonglez)

- 30k con per client VM
- Unbound

TCP is 25% of UPD with 25,000 clients

1 thread
Latest work

• We wanted to **reproduce 25,000 clients** for Unbound on our setup
  • Plus… Don’t restrict nameserver to 1 thread

• Extend to **other nameservers** since Unbound doesn’t do concurrent processing

• And….Baptiste is doing similar measurements!
Latest results - Unbound

25,000 clients
5000 q/con

[Graph: Unbound UDP vs TCP (1 thread)]

Uni-directional line.
Latest results - Unbound

- 25,000 clients
- 5000 q/con
- 1 thread
- 160 kqps
- TCP ~32%

Graph: Unbound UDP vs TCP (1 thread)
Latest results - Unbound

- 25,000 clients
- 5000 q/con
- 1 thread
- 160 kqps
- TCP ~32%
Latest results - Unbound

- **1 thread**
  - 160 kqps
  - TCP ~32%

- **32 threads**
  - 620 kqps
  - TCP ~67%

- 25,000 clients
- 5000 q/con

### Graphs

- **Unbound UDP vs TCP (1 thread)**

- **Unbound UDP vs TCP (32 threads)**
Latest results - BIND & Knot R

25,000 clients
5000 q/con

Knot Resolver UDP vs TCP (32 threads)
Latest results - BIND & Knot R

25,000 clients
5000 q/con

Knot Resolver UDP vs TCP (32 threads)

UDP
TCP

600 kqps
TCP ~50 %
Latest results - BIND & Knot R

Knot Resolver UDP vs TCP (32 threads)

- UDP
- TCP

BIND UDP vs TCP (32 threads)

- UDP
- TCP

25,000 clients
5000 q/con

600 kqps
TCP ~50 %
Latest results - BIND & Knot R

Knot Resolver UDP vs TCP (32 threads)

- 600 kqps
- TCP ~50%

BIND UDP vs TCP (32 threads)

- 310 kqps
- TCP ~25%

25,000 clients
5000 q/con
MY WORK HERE IS DONE

RIGHT?
Latest results - Unbound

25,000 clients
5000 q/con

Unbound UDP vs TCP (32 theads)
Latest results - Unbound

25,000 clients
5000 q/con

TCP 410 kqps

Unbound UDP vs TCP (32 theads)
Latest results - Unbound

25,000 clients
5000 q/con

Unbound UDP vs TCP (32 threads)

TCP 410 qcps

Unbound (32 threads) - QPS per client
Latest results - Unbound

25,000 clients
5000 q/con

Unbound UDP vs TCP (32 threads)

TCP 410 kqps

Client View:
~ 20 qps per client
Reality check - uniform TCP client traffic isn’t real!

- **UDP** benchmarking can get away with few uniform clients.

- Session based benchmarking can’t do this:
  - Consider individual client experience (throughput & latency)
  - Clients behave differently: must simulate client population with varying profiles (qps, idle timeouts)
Reality check - uniform TCP client traffic isn’t real!

- **UDP** benchmarking can get away with few **uniform clients**

- Session based benchmarking can’t do this:
  - Consider individual client experience (**throughput** & latency)
  - Clients behave differently: must **simulate client population** with varying profiles (qps, idle timeouts)

This is a typical approach of HTTP benchmarking software, but very little data for DNS
Sample client profiles

Trimodal distribution of queries per client

Number of clients

Number of queries

7 minutes (420s) of traffic
~250k clients

Courtesy of Bert Hubert, PowerDNS
Sample client profiles

Trimodal distribution of queries per client

- Number of clients
- Number of queries

- 20 qps
- 7 minutes (420s) of traffic
- ~250k clients

Courtesy of Bert Hubert, PowerDNS
Sample client profiles

Trimodal distribution of queries per client

- 0.2 qps
- 1 q every ~5 s
- 20 qps

Courtesy of Bert Hubert, PowerDNS

7 minutes (420s) of traffic ~250k clients
Sample client profiles

7 minutes (420s) of traffic
~250k clients

0.03 qps
1 q every ~35 s

0.2 qps
1 q every ~5 s

20 qps

Courtesy of Bert Hubert, PowerDNS
Sample client profiles

Aggregation doesn’t apply!

Factors involved:
- Client resolver choice
- Provider load balancing
- Software periodic probing
- Bursty traffic/idle time
- Routers with many devices
- Forwarding resolvers

- 0.03 qps
  1 q every ~35 s

- 0.2 qps
  1 q every ~5 s

- 20 qps

7 minutes (420s) of traffic
~250k clients

Courtesy of Bert Hubert, PowerDNS
Re-purposing HTTP load testers for DNS?

- Surveyed many, experimented with two:
  - **k6**: Golang, JavaScript, currently HTTP only
    - **Prohibitive startup times** with 1000 VU
  - **Tsung**: Erlang, supports for non-HTTP protocols, supports client profiles
    - BUT peak traffic generation for **single client** instance 30k clients, **100kqps** (still need several client VMs)
    - Adding sync DNS was easy, but doesn’t pipeline properly

Would like to avoid needing large client farms
Future of DNS BM?

- **Hybrid** tool required: DNS query throughput but with HTTP tester-like scripting of different client session types
  - May still need client VM farms?
- Extendable to DoH, DoQ, foo
- Nothing exists today to do this:
  - Requirements wish list coming - please comment!
  - Anonymised client data - can you collect?
  - Collaborate, contribute code or funds - let us know!
Thank you!

dnsprivacy.org
@DNSPrivacyProject
Test setup - Hardware

- 2*8 core Intel Xenon @ 2.1Ghz, 32Gb RAM
- Ubuntu 18.04
- Only basic OS and NS tuning
- Hot cache of 10M names

'Out of the box' testing