DNS Privacy - Implementation and Deployment Status

Sara Dickinson
Sinodun

Allison Mankin
Salesforce

IEPG@IETF96

Berlin, July 2016

DNS Privacy - Background

- **RFC 7558** - “Pervasive Monitoring is an Attack”
- **DPRIVE WG** (formed in 2014)

Current Charter: Stub to Recursive ONLY

  - **RFC 7626**: DNS Privacy Considerations
  - **RFC 7858:** Specification for DNS over TLS

Port 853 Allocated

- **RFC 7816:** QNAME Minimisation
  - Recursive (Rec) to Authoritative (Auth)

RFC 7626 - DNS Privacy Considerations

Worth a read - many operational issues here!

- Expert coverage of risks throughout DNS ecosystem
- Rebuts “alleged public nature of DNS data”

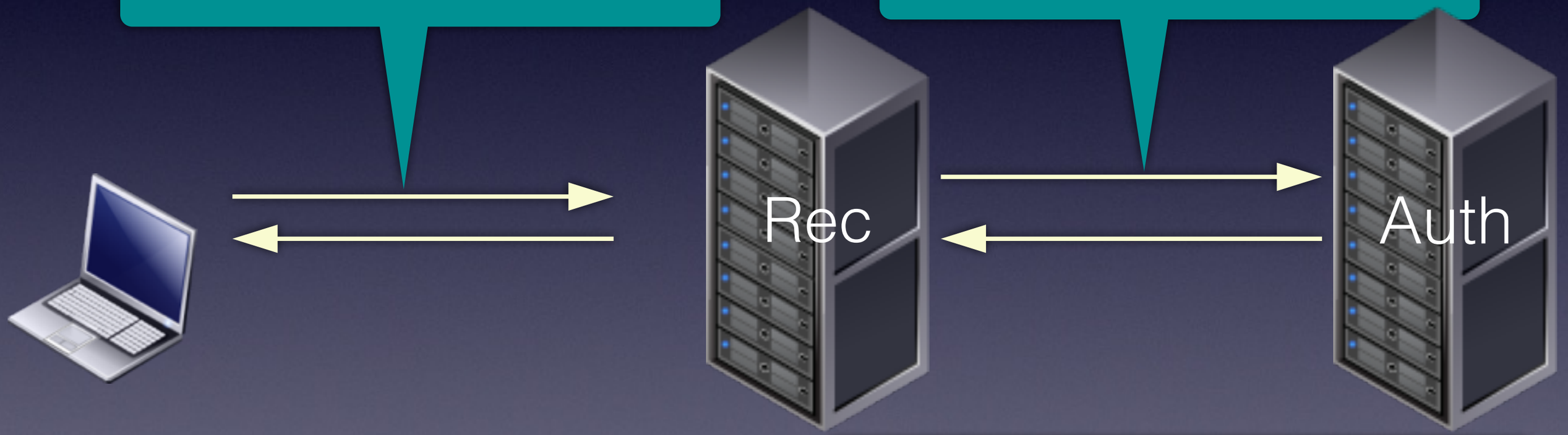
DNS Disclosure Example

? iepg.org

? www.intercontinental.com

? iepg.org

? www.intercontinental.com



Stub Query => Rec
user src address

Client Subnet option (RFC7871)
contains source subnet
in DNS query

DNS Disclosure Example

No QNAME MIN

iepg.org

iepg.org

iepg.org

DNS Disclosure Example

With QNAME MIN

org

Root

iepg.org

Rec

iepg.org

Auth
for .org

Risk Mitigation Matrix

	In-Flight		At Rest	
Risk	**Stub => Rec**	**Rec => Auth**	**At Recursive**	**At Authoritative**
Monitoring (Passive/Active)	**Encryption (DNS-over-TLS)**			
		QNAME Minimization		
Other Disclosure Risks e.g. Data breaches			**Data Best Practices (Policies) e.g. De-identification**	

Operational Challenges

Considerations for Operators

- TLS operation is a new challenge for DNS recursive operators
- Note well: historic DNS servers have **very** basic TCP capabilities
    - Newer software is adding more sophistication and modern TCP features
- In addition, TLS is evolving…

TCP/TLS Scalability

- Historic measurements used 1-shot TCP, gave results significantly worse than UDP and under reported capacity
- New DNS-over-TCP/TLS benchmarking tools are on the way (patch to dnsperf).

Implementation Status

Recursive implementations

Features		Recursive resolver		
		Unbound	BIND	Knot Res
TCP/TLS Features	TCP fast open			
	Process pipelined queries			
	Provide OOOR			
	EDNS0 Keepalive			
TLS Features	TLS on port 853			
	Provide server certificate			
	EDNS0 Padding			
Rec => Auth	QNAME Minimisation			

Dark Green: Latest stable release supports this
Light Green: Patch available
Yellow: Patch/work in progress, or requires building a patched dependency
Grey: Not applicable or not yet planned

Stub implementations

Features		Stub			
		ldns	digit	getdns	BIND
TCP/TLS Features	TCP fast open				
	Connection reuse				
	Pipelining of queries				
	Process OOOR				
	EDNS0 Keepalive				
TLS Features	TLS on port 853				
	Authentication of server				
	EDNS0 Padding				

Dark Green: Latest stable release supports this
Light Green: Patch available
Yellow: Patch/work in progress, or requires building a patched dependancy
Grey: Not applicable or not yet planned

** getdns uses libunbound in recursive mode*

Deployment Status

RECURSIVE

Test Servers

- NLnet Labs have a test server today. Details:

https://portal.sinodun.com/wiki/display/TDNS/DNS-over-TLS+test+servers

- OARC also offering trial servers (members only at the moment).

STUB

- Modern **async DNSSEC** enabled API
    - https://getdnsapi.net
- Stub mode is feature rich for DNS Privacy
    - Alpha (v1.1.0a1) of a daemon mode - try it out:

https://portal.sinodun.com/wiki/display/TDNS/DNS+Privacy+daemon

- Challenge: Adoption in OS
    - nss_switch module?

RECURSIVE

Test Servers

- RIPE DNS WG: Presentation and discussion of offering experimental DNS Privacy Service
- RIPE are planning to co-ordinating a community effort
  - Research various solutions and issues
  - Output will be operational guidance

Summary

- Good reasons to consider DNS Privacy
- Active work on DNS Privacy standards and implementation
- Can test DNS Privacy today using getdns & current test servers
- More DNS Privacy services on the way…

Thank you!

Any Questions?

sara@sinodun.com
amankin@salesforce.com

Additional Slides

DNS-over-TLS needs TCP !

- DNS-over-TCP… historically used only as a fallback transport (TC=1 ➡️ ‘one-shot’ TCP, Zone transfer)
- RFC7766 (2016) - a bis of RFC5699
    - TCP a **requirement** for DNS implementations
    - Performance on par with UDP, security/robustness
- RFC7828 - edns0-tcp-keepalive
    - Timeouts for persistent TCP connections

TCP/TLS Performance

Goals:

1. Optimise TCP/TLS set up & resumption
    - TCP FastOpen, TLS resumption, [TLS 1.3]

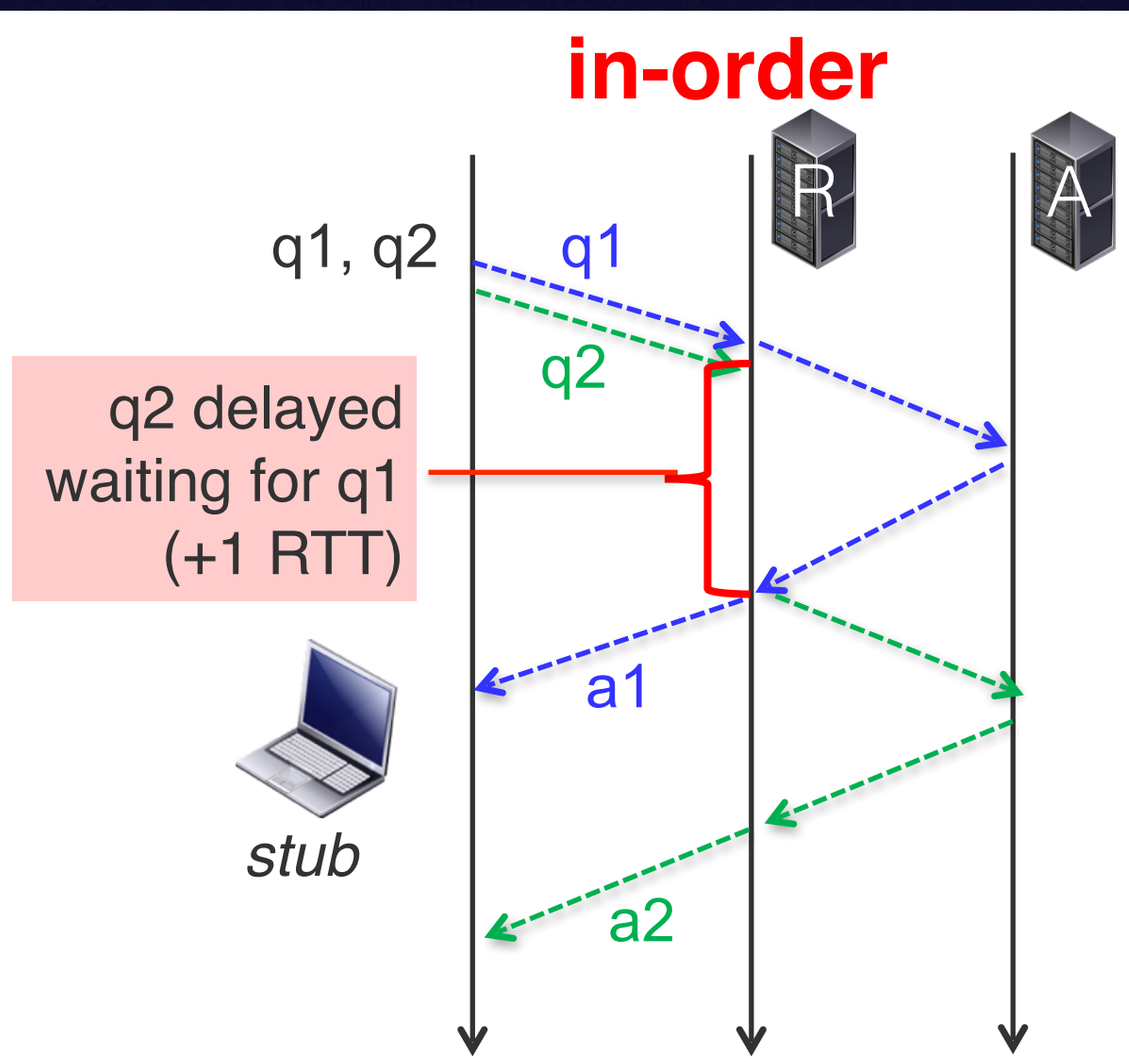
2. Amortise cost of TCP/TLS setup
    - Keep connection open, send many messages efficiently

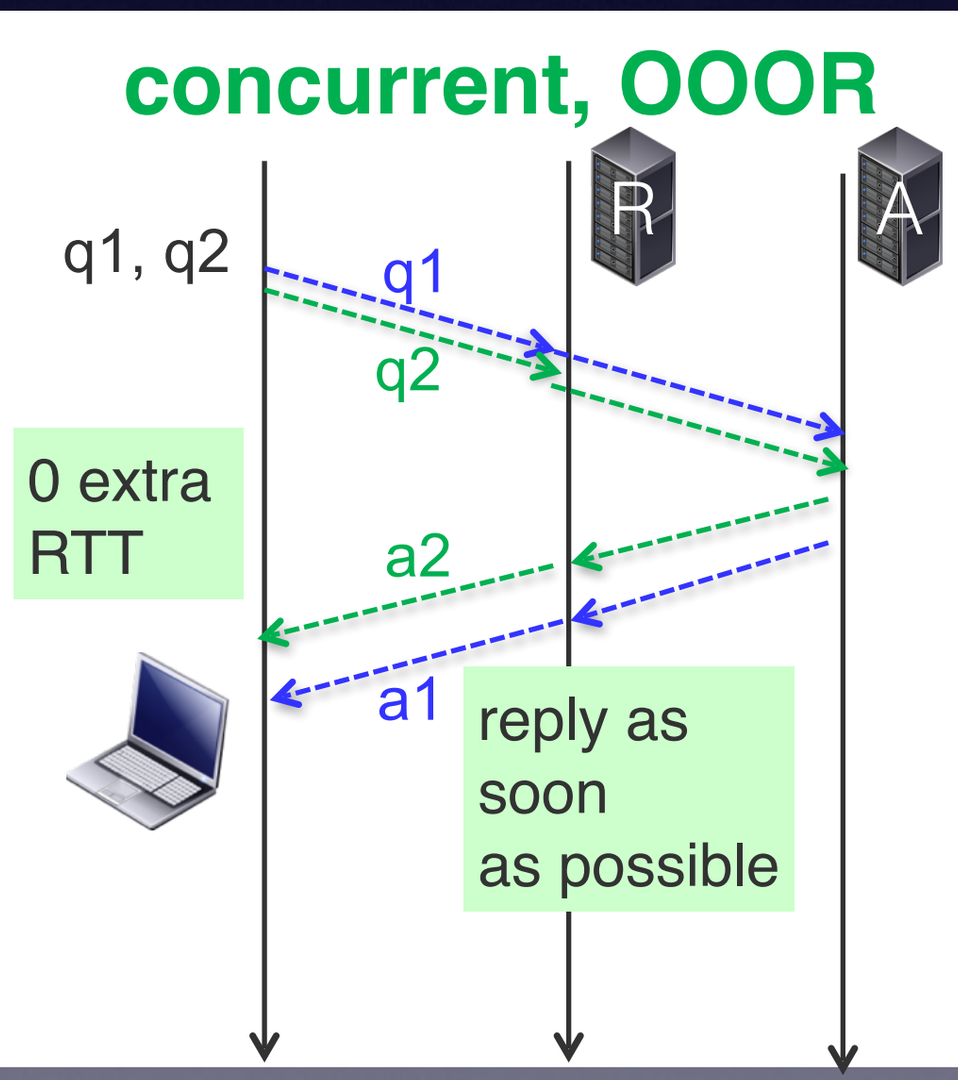
3. Server must handle many connections robustly
    - Learn from HTTP servers

Performance (RFC7766)

Client - pipeline requests and handle out-of-order response

Server - concurrent processing of requests sending of out of order responses





Alternative server side solutions

- dnsdist would be great… but no support yet
- Pure TLS load balancer
  - NGINX
  - BIND article on using stunnel (add link)

Disadvantages

- server must still have full TCP capabilities
- pass through of edn0-tcp-keepalive option
- DNS specific access control is missing

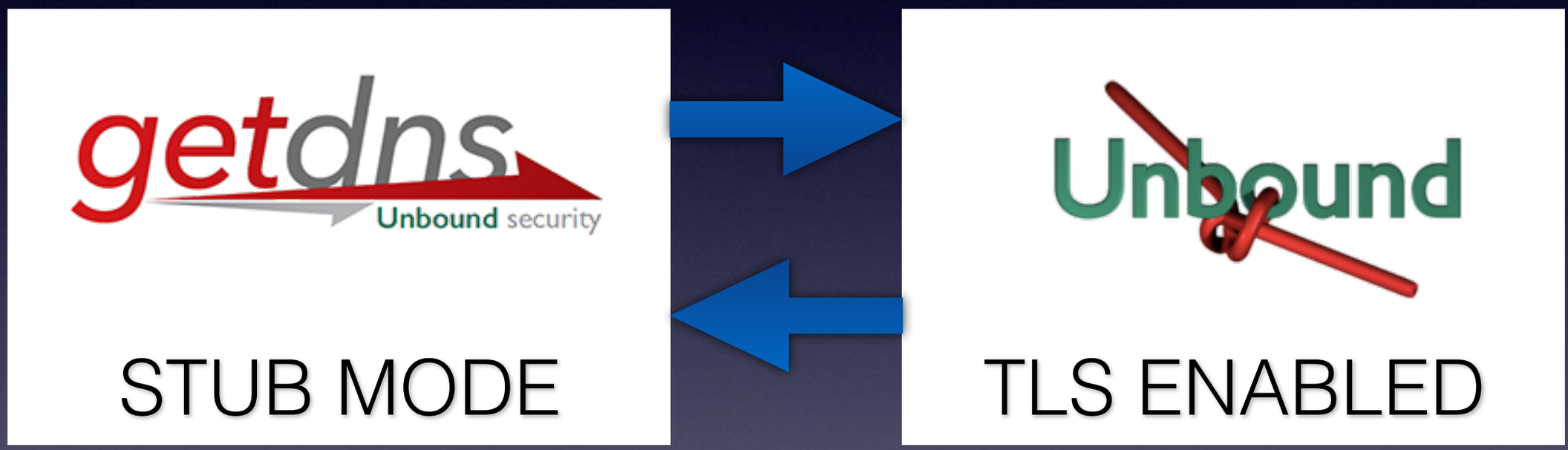
TLS BCP

- UTA (Using TLS in Applications) WG produced RFC7525 this year - “BCP for TLS and DTLS”

DNS-over-TLS is relatively ‘green-field’

- Key recommendations - Protocol versions:
  - **TLS v1.2** MUST be supported and preferred
- Recommended Cipher Suites (4 of ~100):
  - **AEAD mode** - Forward secrecy for key exchange
  - TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256

Examples



1.0.0b1

1.5.9

Scenario 1:

Strict TLS

- Configuration:
  - **Hostname verification required**
    - Correct hostname for Unbound resolver
    - TLS as only transport
- RESULT:
  - TLS used (cert & hostname verified)

Scenario 2:

Strict TLS

- Configuration:
  - Hostname verification required (Default)
  - **No or incorrect hostname**
  - TLS as only transport
- RESULT:
  - Query fails

Scenario 3:

Opportunistic TLS

- Configuration:
  - **Hostname verification optional**
  - Valid, none or incorrect hostname
  - TLS as only transport
- RESULT:
  - TLS used (hostname verification tried but fails)

Scenario 4:

Opportunistic TLS

- Configuration:
  - Hostname verification required (default)
  - Valid, none or incorrect hostname
  - **TLS with fallback to TCP**
- RESULT:
  - TLS used (hostname verification tried but fails)