IETF DPRIVE WG: Encrypting DNS

Sara Dickinson
Sinodun

ICANN 54 - Tech Day
October 2015
DPRIVE WG

Focus is stub to recursive

Group created

IETF91

IETF92

ID: problem-statement

ID: dns-tls-newport

ID: start-tls-for-dns

ID: dns-tls-newport

ID: start-tls-for-dns

STARTTLS & port

ID: start-tls-for-dns

ID: dns-over-tls

Early port allocation

ID: dnsodtls

ID: dnsodtls

RFC7626

2014

Q4

Q1

Q2

Q3

Q4

2015

TIME
# Pros and Cons

<table>
<thead>
<tr>
<th></th>
<th>Pros</th>
<th>Cons</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>STARTTLS</strong></td>
<td>• Port 53</td>
<td>• Port 53 - middleboxes?</td>
</tr>
<tr>
<td></td>
<td>• Known technique</td>
<td>• Existing TCP implementations</td>
</tr>
<tr>
<td></td>
<td>• Incrementation deployment</td>
<td>• Downgrade attack on negotiation</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Latency from negotiation</td>
</tr>
<tr>
<td><strong>TLS (new port)</strong></td>
<td>• New DNS port (no interference with port 53)</td>
<td>• New port assignment</td>
</tr>
<tr>
<td></td>
<td>• Existing implementations</td>
<td></td>
</tr>
<tr>
<td><strong>DTLS</strong></td>
<td>• UDP based</td>
<td>• Truncation of DNS messages (just like UDP)</td>
</tr>
<tr>
<td></td>
<td>• Certain performance aspects</td>
<td>➡ Fallback to clear text or TLS</td>
</tr>
<tr>
<td></td>
<td></td>
<td>❌ Can’t be standalone solution</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• No running code</td>
</tr>
</tbody>
</table>
Early port allocation

- 8th October 2015 - IANA assigned **port 853**:

<table>
<thead>
<tr>
<th>protocol</th>
<th>port</th>
<th>protocol</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>domain-s</td>
<td>853</td>
<td>tcp</td>
<td>DNS query-response protocol run over TLS/DTLS</td>
</tr>
<tr>
<td>domain-s</td>
<td>853</td>
<td>udp</td>
<td>DNS query-response protocol run over TLS/DTLS</td>
</tr>
</tbody>
</table>
DNS-over-TLS needs TCP!

- DNS-over-TCP... historically used only as a fallback transport (TC=1 ➔ ‘one-shot’ TCP, Zone transfer)

- 2010: RFC5966 - TCP a **requirement** for DNS implementations

- 2014: Connection-oriented DNS - USC/ISI paper

- `draft-ietf-dnsop-5966bis`
  - performance on par with UDP, security/robustness

- `draft-ietf-dnsop-edns-tcp-keepalive` - persistent TCP connections
TCP/TLS Performance

Goals:

1. Handle many TCP connections robustly
2. Optimise TCP/TLS set up & resumption
   - TCP FastOpen, TLS resumption, [TLS 1.3]
3. Amortise cost of TCP/TLS setup
   - Send many messages efficiently
Performance (5966bis)

Client - Query pipelining

connection re-use

q1, q2

q2 delayed waiting for q1 (+1 RTT)

(stub)

q1, q2

a1

q2

a2

(recursive)

pipelining

q1, q2

q1

a1

q2

a2

0 extra RTT
Performance (5966bis)

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

- In-order:
  - q1, q2
  - q1
  - q2
  - a1
  - a2
  - q2 delayed waiting for q1 (+1 RTT)

- Concurrent, OOOR:
  - q1, q2
  - q1
  - q2
  - a1
  - a2
  - 0 extra RTT
  - reply as soon as possible

stub
DNS-over-TLS implementations

- **Unbound 1.4.14 (2011)** - DNSTrigger
- **TLS patches** for LDNS and NSD
- [BIND TCP improvements]
- **getdns** - ongoing development of DNS-over-TLS
• Modern **async DNSSEC** enabled API
  
  • https://getdnsapi.net

• Stub mode has TLS with flexible privacy policy and fallback:
  
  ✮ Strict (Authenticated) TLS only
  ✮ Opportunistic TLS
  ✮ Fallback to TCP, UDP

• Pipelining, OOOP, Configurable idle time
## Current status

<table>
<thead>
<tr>
<th>Software</th>
<th>digit</th>
<th>LDNS</th>
<th>getdns</th>
<th>Unbound</th>
<th>NSD</th>
<th>BIND</th>
</tr>
</thead>
<tbody>
<tr>
<td>mode</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>client</td>
<td></td>
<td>client (drill)</td>
<td>stub</td>
<td>recursive*</td>
<td>server</td>
<td>client</td>
</tr>
<tr>
<td>client</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(drill)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>TLS</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>TFO</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Conn reuse</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Pipelining</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>OOOP</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

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

* getdns uses libunbound in recursive mode
UTA (Using TLS in Applications) WG produced RFC7525 this year - “BCP for TLS and DTLS”

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**

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

• Secure discovery of certificate/hostname/etc.
• For DNS-over-TLS?
  • Pre-deployed configuration profile
  • DANE… (clear-text or un-authenticated TLS)
  • boot strap problem
Summary

• Active work on encrypting DNS in DPRIVE
• For DNS-over-TLS performance is key
• Client should consider privacy policy
  • see Appendix for stub/recursive examples
• Know your (D)TLS Best Current Practices
Thank you!

Any Questions?

sara@sinodun.com
Appendix
Examples

- STUB MODE
- TLS ENABLED

Next release: Hostname verification

1.5.5
Scenario 1: Strict TLS

- Configuration:
  - Hostname verification required (Default)
  - 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)
Example

STUB MODE

NO TLS
Scenario 3: Opportunistic TLS

- Configuration:
  - Hostname verification required (default)
  - Valid, none or incorrect hostname
  - TLS with fallback to TCP

- RESULT:
  - TCP used (TLS tried, but fails)