DNS over CoAP (DoC)
draft-lenders-dns-over-coap
(https://datatracker.ietf.org/doc/draft-lenders-dns-over-coap/)

Martine S. Lenders, Christian Amsüss, Cenk Gündoğan,
Thomas C. Schmidt, Matthias Wählisch
IETF 114 CoRE Meeting, 2022-07-26
Outline

Introduction

Changes since interim-2022-core-06

DNS Push and CoAP Observe
Motivation

Attack Scenario

Countermeasure: Encrypt name resolution triggered by IoT devices
Our proposal: DNS over CoAP

• **Encrypted communication** based on DTLS or OSCORE

Additional advantages:

• **Block-wise message transfer** to overcome Path MTU problem
• **Share system resources** with CoAP applications
  • Same socket and buffers can be used
  • Re-use of the CoAP retransmission mechanism
Changes: Reduce Restatements of CoAP behavior

- Be more precise when Confirmable (CON) messages SHOULD be used
- Clean-up paragraph on error handling
  - Removed block-wise recommendations
  - Removed sentence that stated that FETCH is sent to server URI
  - Removed considerations on proxies

Caching:

- Provide algorithm on CoAP Max-Age vs DNS TTL mitigation
- Remove ETag considerations
Changes: Done TBDs

+ Recommend OSCORE usage
+ Draft out Observe usage (needs work, see next slides)
  – Remove TBD for GH Issue #4; Draft on compressed Content-Format planned

Still TBD: IANA Considerations, pick ID for “application/dns-message”
Content-Format
Primer on DNS Push Notifications (RFC 8765)

- Based on DNS Stateful Operations (DSOs, RFC 8490)
- Orthogonal to classic QUERY/RESPONSE paradigm

DNS Client  Recursive Resolver

```
DSO [id,SUBSCRIBE(example.org)]
```

Requires DNS over TLS and additional state information at client
⇒ Use CoAP Observe as signal to use SUBSCRIBE instead of QUERY at DoC Server
Primer on DNS Push Notifications (RFC 8765)

- Based on DNS Stateful Operations (DSOs, RFC 8490)
- Orthogonal to classic QUERY/RESPONSE paradigm

![Diagram](image-url)
Primer on DNS Push Notifications (RFC 8765)

- Based on DNS Stateful Operations (DSOs, RFC 8490)
- Orthogonal to classic QUERY/RESPONSE paradigm

DNS Client

Recursive Resolver

DSO [id, SUBSCRIBE(example.org)]

DSO [id, NOERROR]

DSO [0, PUSH(example.org:2001:db8::1, TTL=300)]

DSO [0, PUSH(example.org:2001:db8::2, TTL=300)]

Requires DNS over TLS and additional state information at client

⇒ Use CoAP Observe as signal to use SUBSCRIBE instead of QUERY at DoC Server
Primer on DNS Push Notifications (RFC 8765)

- Based on DNS Stateful Operations (DSOs, RFC 8490)
- Orthogonal to classic QUERY/RESPONSE paradigm

**Diagram:**

```
DNS Client  Recursive Resolver

DSO [id,SUBSCRIBE(example.org)]

DSO [id,NOERROR]

DSO [0,PUSH(example.org:2001:db8::1,TTL=300)]

DSO [0,PUSH(example.org:2001:db8::2,TTL=300)]

DSO [0,UNSUBSCRIBE(id)]
```
Based on DNS Stateful Operations (DSOs, RFC 8490)
Orthogonal to classic QUERY/RESPONSE paradigm

Requires DNS over TLS and additional state information at client
Primer on DNS Push Notifications (RFC 8765)

- Based on DNS Stateful Operations (DSOs, RFC 8490)
- Orthogonal to classic QUERY/RESPONSE paradigm

**DNS Client**

- DSO `[id, SUBSCRIBE(example.org)]`
- DSO `[id, NOERROR]`
- DSO `[0, PUSH(example.org:2001:db8::1, TTL=300)]`
- DSO `[0, PUSH(example.org:2001:db8::2, TTL=300)]`
- DSO `[0, UNSUBSCRIBE(id)]`

**Recursive Resolver**

Requires DNS over TLS and additional state information at client

⇒ Use CoAP Observe as signal to use SUBSCRIBE instead of QUERY at DoC Server
Proposal: CoAP Observe for DNS Push Subscriptions

DoC Client

FETCH [Obs:0,QUERY(example.org)]

DoC Server

DSO [id, SUBSCRIBE(example.org)]

DSO [id, NOERROR]

[0, PUSH(example.org:2001:db8::1, TTL=300)]

2.05 [Obs: x, RESP(example.org:2001:db8::1, TTL=300)]

DSO [0, PUSH(example.org:2001:db8::2, TTL=300)]

2.05 [Obs: x + 1, RESP(example.org:2001:db8::1, TTL=300)]

FETCH [Obs:1]

DSO [0, UNSUBSCRIBE(id)]

2.05
Proposal: CoAP Observe for DNS Push Subscriptions

- **DoC Client**: FETCH [Obs:0,QUERY(example.org)]
- **DoC Server**: DSO [id,SUBSCRIBE(example.org)]
  - DSO [id,NOERROR]
  - DSO [0,PUSH(example.org:2001:db8::1,TTL=300)]
- **Recursive Resolver**: 2.05 [Obs:x,RESP(example.org:2001:db8::1,TTL=300)]
  - DSO [0,PUSH(example.org:2001:db8::2,TTL=300)]
  - 2.05 [Obs:x+1,RESP(example.org:2001:db8::1,TTL=300)]
  - FETCH [Obs:1]
  - DSO [0,UNSUBSCRIBE(id)]
Proposal: CoAP Observe for DNS Push Subscriptions

DoC Client

FETCH [Obs:0,QUERY(example.org)]

2.05 [Obs:x,RESP(example.org:2001:db8::1,TTL=300)]

2.05 [Obs:x + 1,RESP(example.org:2001:db8::1,TTL=300)]

DoC Server

DSO [id,SUBSCRIBE(example.org)]

DSO [id,NOERROR]

[0,PUSH(example.org:2001:db8::1,TTL=300)]

[0,PUSH(example.org:2001:db8::2,TTL=300)]

Recursive Resolver

[0,UNSUBSCRIBE(id)]

2.05
Proposal: CoAP Observe for DNS Push Subscriptions

**DoC Client**

```
FETCH [Obs:0, QUERY(example.org)]
```

```
2.05 [Obs:x, RESP(example.org:2001:db8::1, TTL=300)]
```

```
2.05 [Obs:x + 1, RESP(example.org:2001:db8::1, TTL=300)]
```

```
FETCH [Obs:1]
```

```
2.05
```

**DoC Server**

```
DSO [id, SUBSCRIBE(example.org)]
```

```
DSO [id, NOERROR]
```

```
[0, PUSH(example.org:2001:db8::1, TTL=300)]
```

```
[0, PUSH(example.org:2001:db8::2, TTL=300)]
```

```
DSO [0, UNSUBSCRIBE(id)]
```

**Recursive Resolver**

```
```
Use Case Example: Subscribe to list of services using DNS-SD (Subscribe)

FETCH coap://[2001:db8::1]/dns
Observe: 0
---
QUERY ID: 0
Questions:
- qname=_coap._udp.local,qtype=PTR,qclass=IN
Use Case Example: Subscribe to list of services using DNS-SD (Push)

2.05 Content
Observe: 2060
Max-Age: 3600
---
RESPONSE ID: 0
Questions:
- qname=_coap._udp.local,qtype=PTR,qclass=IN
Answers:
- name=0xc00c(_coap._udp.local),type=PTR,class=IN,ttl=3600,
  rdata=_dns._coap._udp.local
- name=0xc00c(_coap._udp.local),type=PTR,class=IN,ttl=3600,
  rdata=_lamp1.coap._udp.local