DNS privacy

藤原 和典
<fujiwara@jprs.co.jp>
株式会社日本レジストリサービス (JPRS)
2015/11/19
(2015/12/28 追記)

自己紹介

- 氏名:　藤原和典
- 個人ページ: http://member.wide.ad.jp/~fujiwara/
- 勤務先:　株式会社日本レジストリサービス (JPRS)
  技術研究部
- 業務内容:DNS関連の研究・開発
- IETFでの活動 (2004~)
  - RFC 5483 6116 (2004~2011):ENUMプロトコル
  - RFC 5504 5825 6856 6857 (2005~2013)
    - メールアドレスの国際化 (互換性部分を担当)
  - draft-fujiwara-dnsop-ds-query-increase(2013/6~)
  - draft-fujiwara-dnsop-poisoning-measures (2014/7)
  - draft-ietf-dnsop-dns-terminology (2014/11～)
  - draft-fujiwara-dnsop-nsec-aggressiveuse (2015/3~)

用語:フルリゾルバ (Full-service Resolver)

- キャッシュサーバという言葉は曖昧であるため、本資料ではフルリゾルバ / Full-service Resolverを使用する
  - PowerDNSやBundy権威DNSサーバは権威DNSサーバだがキャッシュをサービスに使用
  - RFC 1035では名前解決する機能を持つものをresolverとし、full resolver、resolver+cache、Recursive server+Central cacheの三通りの書き方で説明
  - RFC 1123ではstub resolverとfull-service resolverを区別して紹介
  - Full-service resolverかFull resolverが正式名
  - 他のRFCではCaching serverという言葉を定義なしに使用 (Cache serverは使用されていない)
  - draft-ietf-dnsop-dns-terminology 参照

政府による盗聴問題

- 政府による盗聴問題
  - Great Firewall による検閲
  - エドワード･スノーデン氏による告発(2013/5)
- 二大国関連の不安
  - 滞在中の通信傍受？
  - 企業は政府にデータ提供？ (スノーデン氏の資料)
    - Public DNS, 検索, ストレージ, A*, i*, W*
  - NIST暗号にバックドア？
    - FIPS-* : AES, SHA*, ECDSA P-*, RSAパラメータ

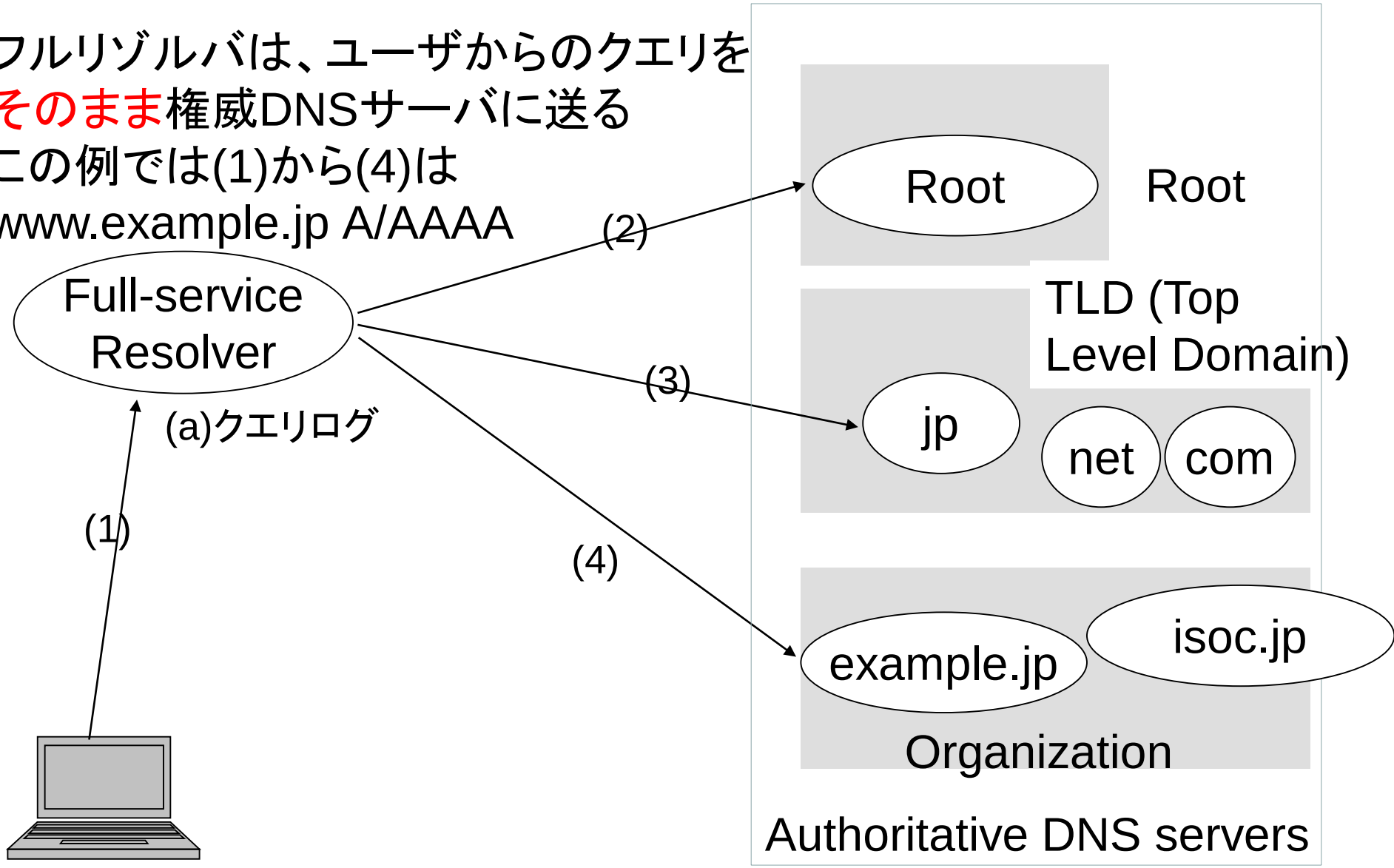
  → 日本企業のサービスを使う？　割り切る？ VPN？
- 日本国での不安
  - 「通信の秘密」はよく担保されている？
  - (1986年　日本共産党幹部宅盗聴事件)
  - 犯罪捜査のための通信傍受に関する法律
  - ちょっと不安 → できるだけ守る (global standard)

本日のテーマ: DNS Privacy

- DNSのプライバシー？
  - IPアドレス(個人?)の通信情報の秘匿性？
  - IPアドレス(個人?)が何に関心があるかわかるのは問題
- 権威DNSサーバに漏れる情報
  - ルートやTLDでは、細かいホスト名・クエリタイプが見える
- このあとの構成
  - IETFでの問題提起
  - DNS通信路の暗号化
  - クエリ情報漏洩の最小化
  - 今後の見通し

復習:DNSの動作とクエリ情報

フルリゾルバは、ユーザからのクエリを
そのまま権威DNSサーバに送る
この例では(1)から(4)は
www.example.jp A/AAAA



復習:DNSクエリが持つ情報

取れるデータ
フルリゾルバのIPアドレス
時刻、クエリ名、クエリタイプ
ある組織/ISPのユーザが、いつ、なにを見ようとしたかがわかる
キャッシュにより取れるデータは限られる

例: 以下のような名前が漏れている
_bittrrent-tracker._tcp.example.jp SRV
_kerberos._tcp.dc._msdcs.xx.example.jp SRV

Root
Root
クエリログ
クエリ情報収集

TLD (Top Level Domain)

Full-service Resolver

(3)
タッピング

jp
net
com
クエリログ
クエリ情報収集

クエリログ
クエリ情報収集

(1)
タッピング

(4)

例: 以下のような名前が漏れている
時刻 192.0.2.2 www.nic.ad.jp AAAA
時刻 198.51.100.3 www.google.com A
時刻 203.0.113.4 _443._tcp.interop.jp TLSA

取れるデータ
クライアントのIPアドレス
時刻、クエリ名、クエリタイプ
だれ(IPアドレス)が、いつ、なにを見ようとしたかがわかる

Organization

thoritative DNS servers

(0)enter http://www.example.jp/ into browser

参考: DNSクエリ情報収集活動

- Root
  - 年に一度、50時間、ルートサーバなどのクエリ情報を収集
  - 研究やRootの運用に使用:name collisionの評価 (ICANN)
- JP
  - Rootと同じタイミングなどで、全JP DNSのクエリ情報を収集
  - [AG].DNS.JPクエリログを2004年から継続して収集
  - 研究やJPの運用に使用
- フルリゾルバ
  - 大学などで、組織内向けに提供しているフルリゾルバのクエリ情報を収集し、研究に使用
  - Google Public DNS
    - IPアドレスだけ集め、24時間で消すと以下に書かれている
    - https://developers.google.com/speed/public-dns/faq?hl=ja#privacy

IETFでの問題提起

- スノーデン氏による告発後のIETF会議が、アヤシイ熱気に包まれていた (2013/11, IETF 88)
  - 「ワレワレハ・・・・・・」　「オーーーーー」
    - » (政治団体/新興宗教かとおもったのは内緒)
  - できるところから、通信の暗号化を行うという提案
  - 2013/10 NANOG 59でもスノーデン氏が使用していたメールシステム(Lavabit)提供者が絶賛されていた
- RFC 7258 (2014/5発行)
  - Pervasive Monitoring (PM) Is an Attack
  - 広範な監視はプライバシーへの攻撃である
  - IETFはプロトコル設計時にできる範囲で対策するべきである
    → 暗号化や情報の最小化
- 2014/11 IAB Statement on Internet Confidentiality
  - Newly designed protocols should prefer encryption to cleartext operation

IETF DNS関連WGでの対応

- dnsext (プロトコル拡張)
  - 2013/7 完了済
- dnsop (DNS運用ガイドラインの作成)
  - dnsextの機能を引き継ぎ、PM対策の議論を引き受け
  - その後、dprive WGを設立
- dane (DNS(SEC)にTLSの証明書を載せる)
  - もともと暗号化を目的としているため、変化なし
  - 証明書をDNSに載せるTLSA RRの標準化:済
  - SMIMEA, OPENPGPKEY RRの標準化:中

dnsop WGでの議論

- メーリングリストの議論などで、主に二つの改善項目にまとまる
- 対応しないこと (前提)
  - フルサービスリゾルバの管理者は信用する
    - ISPの提供するフルリゾルバや、Public DNSからの情報漏洩については考えない
  - フルサービスリゾルバのアドレスは漏れてよい
- 対応すること
  1. ユーザ端末からフルリゾルバの通信を暗号化
     - なにを見ようとしたかという情報をPMから隠す
  2. フルリゾルバから権威DNSサーバへの情報を最小化 (クエリ情報漏洩の最小化)
     - ルート、TLDオペレータから、ホスト名・クエリタイプを隠蔽

DNSでの改善点

取れるデータ
フルリゾルバのIPアドレス
時刻、~~クエリ名、クエリタイプ~~
ある組織/ISPのユーザが、いつ、~~なにを~~
見ようとしたかがわかる
キャッシュにより取れるデータは限られる

漏れる情報がexample.jp NSだけになる
~~_bittrrent-tracker._tcp~~.example.jp ~~SRV~~ NS
~~_kerberos._tcp.dc._msdcs.xx~~.example.jp ~~SRV~~

Full-service Resolver

Root
Root
対策
クエリログ
クエリ情報収集

TLD (Top Level Domain)

(3)
対策
タッピング
jp
クエリログ
クエリ情報収集
net
com

クエリログ
クエリ情報収集
(1)
タッピング
信用
(4)
対策

漏れる情報が時刻と通信相手だけになる
時刻 192.0.2.2 ~~www.nic.ad.jp AAAA~~
時刻 198.51.100.3 ~~www.google.com A~~
時刻 203.0.113.4 ~~_443._tcp.interop.jp TLSA~~

取れるデータ
クライアントのIPアドレス
時刻、~~クエリ名、クエリタイプ~~
だれ(IPアドレス)が、いつ、~~なにを~~
見ようとしたかがわかる

Organization

thoritative DNS servers

(0)enter http://www.example.jp/ into browser

RFC 7626: DNS Privacy Considerations

- 2015/8 発行
- DNSプライバシーに関する考慮点(リスク分析と攻撃)の提示
- リスク分析
  - “the data in the DNS is public”
  - クエリ名とソースIPアドレスがプライバシー問題と定義
    - RootやTLDでも細かいクエリ名が見える ことを問題視
  - キャッシュからの情報漏洩 (Recursion Desired 0 クエリ)
  - ワイヤタッピング
    - DNSには暗号化の仕組みがない
    - タッピングによるデータ収集はプライバシーへの攻撃
  - サーバでの情報収集: 悪い意思を持つサーバの存在
- 実際の攻撃
  - 肯定的･否定的両方の各種情報収集活動

dprive WG

- DNS PRIVate Exchange (dprive) WG設立
  - 2014年10月17日に設立承認
  - Chairs
    - Tim Wicinski (dnsop WG chair)
    - Warren Kumari (dane WG chair, IEPG chair)
  - スタブリゾルバとフルリゾルバの間の通信をTLS(Transport Layer Security)で暗号化する
  - フルリゾルバと権威DNSサーバの間のプロトコルの変更はしない

DNS通信路の暗号化提案

- IETFで標準化したプロトコルを使って暗号化
  - TLS (Transport Layer Security) (httpsで使用)
  - DTLS(Datagram Transport Layer Security)
    - TLSの機能をUDPで使えるようにしたもの, RFC 6347
- 具体的な案
  - TCPのDNSクエリをTLSで暗号化
    - ポート53を使用してSTARTTLSの仕組みを作成
    - 別のポート番号を使用してTLSをそのまま使用
  - UDPのDNSクエリをDTLSで暗号化
  - 独自暗号の使用
  - 独自プロトコル (DNSCurve) をIETFで採用
- 議論の結果、別ポート番号を使用し、TLSとDTLSを使用する方式の標準化が進展

DNS over TLS

draft-ietf-dprive-dns-over-tls

- 概要
  - TCP port 853 で待ちうけ、(httpsのように)TLS処理
  - DNS over TCP のデータをTLS上に流す
    - 2オクテットのデータ長 + UDP DNSパケットと同じもの
  - TLS/TCPの接続を切らず、張りっぱなしで複数のクエリを処理すること
  - サーバの認証については現在は2種類
    - Opportunistic(認証しない) と 事前設定
- ステータス
  - 2015/10/22~11/12 Working group last call
  - IETF 94でサーバの認証部分を分離することとなったため、若干遅れそうだが、方向性は合意された
  - 追記: 2015/12/7に発行された-02で、サーバの認証プロファイルの追加は他のドキュメントを参照するという記述が追加
  - 追記: 2015/12/9~12/21　二度目のWorking group last call

DNS over TLS　実装

- Unbound: フルリゾルバ
  - Version 1.5.4 から
  - 設定方法 (マニュアルより)
    - ssl-port: 853
    - ssl-service-key: <file>　　TLS秘密鍵ファイル
    - ssl-service-pem: <file>　　TLS公開鍵ファイル
    - ssl-upstream: no　Forwarder動作時にTLSで接続
- getdns api: DNSクライアントライブラリ
  - Version 0.3.0 で入ったようにみえる
- その他パッチあり
- TLS(SSL)アクセラレータと既存実装でも実現可能

クエリ情報漏洩の最小化 (1)

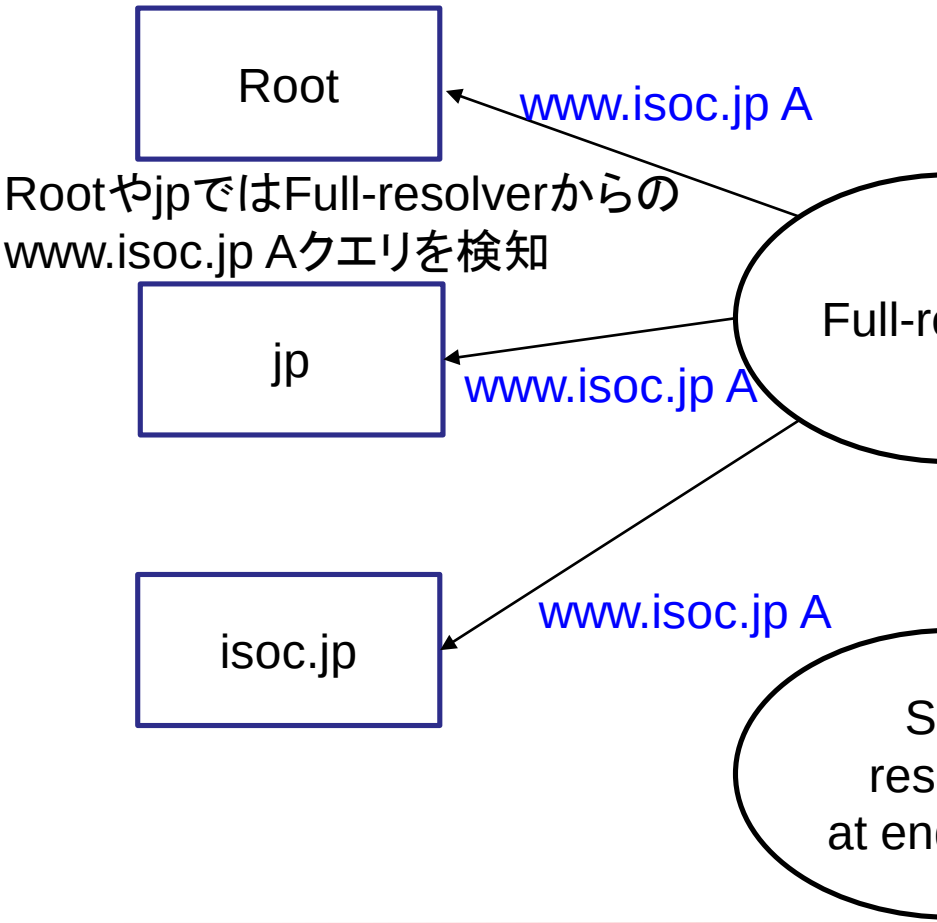
draft-ietf-dnsop-qname-minimisation-07

- プライバシ向上のため、クエリ情報の漏洩を最小化
- 現在のフルリゾルバはユーザからのクエリ名、タイプをそのままルートを含む権威DNSサーバに送る
  - ルート、TLDでユーザからのクエリ名、タイプが見える
  - (キャッシュ効果により、すべてではない)
- そこで、例えばwww.isoc.jp Aを知りたいときに
  - ルートには、TLDのNSクエリ (jp NS)
  - TLDには、登録ドメイン名のNSクエリ (isoc.jp NS)

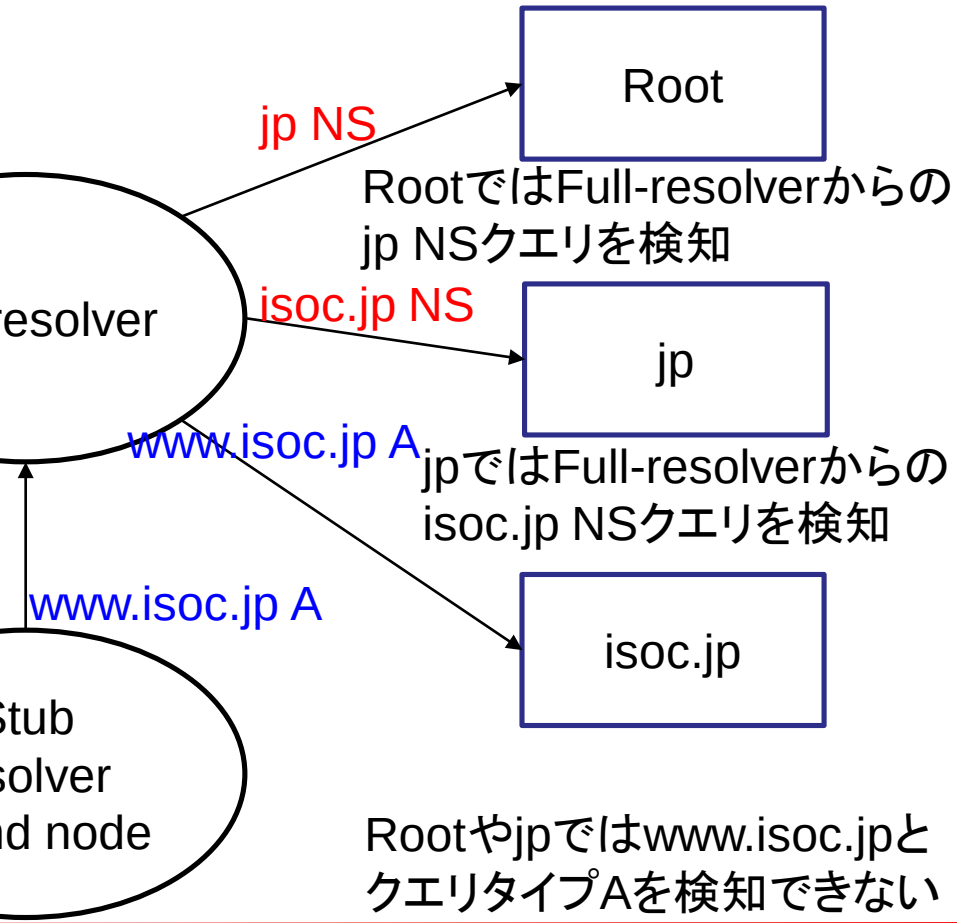
  を送ると、ルート・TLDでもとのクエリが見えない
  - クエリ名 www.isoc.jp, クエリタイプAを隠蔽

クエリ情報漏洩の最小化 (2): 例

従来の動作
同じqname qtype



提案手法
最小限の情報



クエリ情報漏洩の最小化 (3)

- 議論
  - プロトコル違反ではないことの確認
  - アルゴリズムなどの詳細が追加された
  - RFC非準拠な権威DNSサーバを使用するドメイン名の名前解決失敗の可能性指摘
    - ロードバランサなど独自実装の権威DNSサーバに見られる
  - パフォーマンスの懸念
- 現状
  - dnsop WGでの議論は完了、2015/10/12にIESGに提出
  - Experimental: 実験的プロトコルとして標準化見込み
  - 追記: 2015/12/17 IESGから修正点指示 (直せばRFC Editorへ)
- 実装
  - Knot DNS Resolverで実装され、標準で有効化
    - https://gitlab.labs.nic.cz/knot/resolver
    - 現在ベータテスト中で、近いうちにリリース見込みのアナウンス
      - 期待したが、まだConfigureのような仕組みがなく、開発環境と異なる環境で動作させるのは困難そうである

関連する標準化

- dnsop WGでは、TCPでのDNS通信の性能改善が議論され、WGでの議論はほぼ完了
- draft-ietf-dnsop-5966bis-03
  - TCPでのDNS通信についての実装要求仕様
  - パフォーマンス改善のために以下の機能を追加
    - 接続の再利用
    - 複数クエリの同時処理
    - 複数のクエリの順不同応答
    - タイムアウトや切断の規定
    - TCP Fast Open
- draft-ietf-dnsop-edns-tcp-keepalive-04
  - TCPタイムアウト時間を指定するEDNS0オプション

今後必要な標準化

- DNS over DTLS
  - Fragmentationで時間がかかりそう
- DNS over TLS/DTLSのSecurity profile
  - 既存: Opportunistic (検証しないもの)
    - 証明書そのものを検証しない、ホスト名を確認しない
  - 既存: 事前設定
    - 契約書に書いてユーザが入力？
    - /etc/resolv.conf
    - ネットワークのプロパティ？
  - 別のプロトコルでSecurity profile情報を伝える
    - DHCP Option？
    - PPP Option？

今後の見通し

- プロトコル標準化
  - クエリ情報漏洩の最小化: 数ヶ月で完了する見込み
  - DNS over TLS: 数ヶ月で完了する見込み
- フルリゾルバソフトウェアでの実装
  - 標準化が完了するたびに、主要な実装が対応する
  - 複数の実装の開発者・関係者が標準化に深く関与
- クライアント側 (DNS over TLS)
  - getdnsapi は対応済
  - libc の変更には時間がかかるので簡単には使えない
  - アプリケーションごとの対応が進む可能性あり(ブラウザ)
  - FreeBSDのlocal forwarderはUnboundなので実装容易
- Public DNS service
  - 標準化完了後、すみやかに対応するところもありそう

参考資料

- IETF
  - www.ietf.org : RFC, draft, 議事録, アーカイブ
- DNS over TLS　実装
  - www.unbound.net
  - getdnsapi.net
- クエリ情報漏洩の最小化　実装
  - https://gitlab.labs.nic.cz/knot/resolver
    Knot DNS Resolver