DNSプロトコルの進化 2020
(IETFでの標準化)

藤原 和典
fujiwara@jprs.co.jp
株式会社日本レジストリサービス (JPRS)
Internet Week 2020, DNS Day
2020年11月27日

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 (具体的には標準化、試作、論文)
  - Internet Week プログラム委員 (2016～)
- IETFでの活動 (2004~)
  - ENUMプロトコル: RFC 5483 6116 (2004~2011)
  - メールアドレスの国際化 :RFC 5504 5825 6856 6857 (2005~2013)
  - DNS関連の問題提起など
    - RFC 7719, 8499: DNS Terminology → rfc8499bis
    - RFC 8198: DNSSECを用いた名前解決の性能向上
    - draft-ietf-dnsop-avoid-fragmentation: DNSでIP断片化を避ける提案
    - draft-fujiwara-dnsop-delegation-information-signer: 委任情報への署名提案

IANA: DNS Parameters: RR Types (1)

https://www.iana.org/assignments/dns-parameters/dns-parameters.xhtml

TYPE	Value	Meaning	概要
A	1	a host address	DNS
NS	2	an authoritative name server	DNS
CNAME	5	the canonical name for an alias	DNS
SOA	6	marks the start of a zone of authority	DNS
PTR	12	a domain name pointer	DNS
MX	15	mail exchange	DNS
TXT	16	text strings	DNS
AAAA	28	IP6 Address	IPv6
SRV	33	Server Selection	DNS SRV
NAPTR	35	Naming Authority Pointer	[RFC3403]
CERT	37	CERT	[RFC4398]
DNAME	39	DNAME	[RFC6672]

IANA: DNS Parameters: RR Types (2)

TYPE	Value	Meaning	概要
OPT	41	OPT	EDNS
DS	43	Delegation Signer	DNSSEC
SSHFP	44	SSH Key Fingerprint	[RFC4255]
IPSECKEY	45	IPSECKEY	[RFC4025]
RRSIG	46	RRSIG	DNSSEC
NSEC	47	NSEC	DNSSEC
DNSKEY	48	DNSKEY	DNSSEC
DHCID	49	DHCID	[RFC4701]
NSEC3	50	NSEC3	DNSSEC
NSEC3PARAM	51	NSEC3PARAM	DNSSEC

SSHFP

man ssh によるとssh-keygen -r host.example.com.
でSSHFPリソースレコードを作ってくれる
クライアントでの検証は、ssh -o "VerifyHostKeyDNS ask"
(DNSSECで守る方がいいデータ)

IANA: DNS Parameters: RR Types (3)

TYPE	Value	Meaning	概要	Registration Date
TLSA	52	TLSA	DANE	
SMIMEA	53	S/MIME cert association	DANE	2015/12/1
CDS	59	Child DS	DNSSEC自動化	2011/6/6
CDNSKEY	60	DNSKEY(s) the Child wants reflected in DS	DNSSEC自動化	2014/6/16
OPENPGPKEY	61	OpenPGP Key	DANE	2014/8/12
CSYNC	62	Child-To-Parent Synchronization	DNSSEC自動化	2015/1/27
ZONEMD	**63**	**message digest for DNS zone**	**最近の拡張提案**	**2018/12/12**
SVCB	**64**	**Service Binding**	**最近の拡張提案**	**2020/6/30**
HTTPS	**65**	**HTTPS Binding**	**最近の拡張提案**	**2020/6/30**
SPF	**99**	**廃止**	[RFC7208]	**2014/4 廃止**
CAA	257	Certification Authority Restriction	[RFC8659]	2011/4/7

本日の概要

- Internet Week 2016 DNS DAYにて最近のIETF事情を紹介し、2017年にDNS Privacy, 2019年にDNSプロトコルの変化を紹介した。
- 本日は、これまで紹介していないことを中心に、2016年からの変化をまとめる。

- IETFの概要
- DNSを取り扱っているワーキンググループ
- 新しい標準
- 新しい利用例 (Webサーバ、ブラウザでの使用)
- Multicast DNSの拡張

IETF

- IETF (Internet Engineering Task Force)
  - インターネット標準(RFCなど)を決める団体
- IETFの活動への参加(貢献)
  - ドキュメントを書くこと
  - メーリングリストにメールを書くこと
  - 年三回開催される会議に参加することなど: 2020年はすべて遠隔
  - だれでも参加可能
    - 会議への参加費は必要
    - メーリングリストへの参加、標準化提案には費用はかからない
  - 原則として個人での参加
- IETFの活動は公開原則
  - メーリングリスト、会議の議事録、音声

DNS関連WG

- dnsext (DNS Extensions) WG
  - DNSプロトコルの拡張
  - 2013年7月に完了、プロトコル拡張機能をdnsopへ
- dnsop (DNS Operations) WG
  - DNS運用ガイドライン作成
  - DNSプロトコル拡張を作る機能
  - 1999年以前に設立
- dprive (DNS Privacte Exchange) WG
  - DNS通信路を暗号化
  - 2014年10月設立
  - 2016年5月にRFC 7858 DNS over TLS (DoT) を発行
- dane (DNS-based Authentication of Named Entities) WG
  - DNS(SEC)にTLSの証明書を載せる
  - 2010年10月設立、2017年3月完了
- dnssd (Extensions for Scalable DNS Service Discovery) WG
  - .localを使用するMulticast DNS (RFC 6762), DNS-SD (RFC 6763)の拡張
  - 2013年10月設立
- doh (DNS over HTTPS) WG
  - 2017年10月設立 DoHの標準化を目的
  - 2018年10月にRFC 8484: DNS Queries over HTTPS (DoH)発行
  - 2020年3月完了、続く議論をadd WGへ
- add (Adaptive DNS Discovery) WG
  - DNSクライアントがDoT, DoHサーバを見つける方法を定義する
  - 2020年3月設立

赤字は完了したWG　青字は変化

dnsop (DNS Operations) WG

- DNS運用ガイドラインを作るWG
  - DNSプロトコル拡張を作る機能
  - dprive WGはdnsop WGから独立
  - 唯一のDNSそのものを扱うWGとして、ドメイン名全般、DNSプロトコルの話題に関して、IESG, IABなどから意見を求められる
  - 多数の提案を取り扱っている
  - RFCを着実に発行中
    - 2016年1月～2020年11月で28本/5年
    - RFC Editor queueに1本
    - IESG対応中1本
    - WG draft 16本
- 最近のdnsop WGでのテーマ
  - 標準の明確化と修正
    - DNS用語
    - クエリ名情報漏洩の最小化
    - DNSSECアルゴリズム
    - Serve stale
  - 新しい要求
    - ブラウザがHTTPSサーバを見つけるための、新しいリソースレコード (SVCB, HTTPS)
  - セキュリティ対策
    - ゾーン情報のダイジェスト
    - 委任情報確認の厳格化
    - Delegation only
    - DNS Cookies
    - IP断片化回避

dnsop WG: 2016～2020年のRFC (1/2)

- RFC 7766, 2016/3/3: DNS Transport over TCP - Implementation Requirements
- RFC 7816, 2016/3/22: DNS Query Name Minimisation to Improve Privacy
- RFC 7828, 2016/4/6: The edns-tcp-keepalive EDNS0 Option
- RFC 7871, 2016/5/20: Client Subnet in DNS Queries
- RFC 7873, 2016/5/27: Domain Name System (DNS) Cookies
- RFC 7901, 2016/6/21: CHAIN Query Requests in DNS
- RFC 8020, 2016/11/8:NXDOMAIN: There Really Is Nothing Underneath
- RFC 8027, 2016/11/28: DNSSEC Roadblock Avoidance
- RFC 8078, 2017/3/10: Managing DS Records from the Parent via CDS/CDNSKEY
- RFC 8109, 2017/3/15: Initializing a DNS Resolver with Priming Queries
- RFC 8145, 2017/4/15: Signaling Trust Anchor Knowledge in DNS Security Extensions (DNSSEC)
- RFC 8198, 2017/7/25: Aggressive Use of DNSSEC-Validated Cache
- RFC 8244, 2017/10/19: Special-Use Domain Names Problem Statement
- RFC 8501, 2018/11/28: Reverse DNS in IPv6 for Internet Service Providers
- RFC 8509, 2018/12/18: A Root Key Trust Anchor Sentinel for DNSSEC

dnsop WG: 2016～2020年のRFC (2/2)

- RFC 8482, 2019/1/10: Providing Minimal-Sized Responses to DNS Queries That Have QTYPE=ANY
- RFC 8490, 2019/3/15: DNS Stateful Operations
- RFC 8499, 2019/1/2: DNS Terminology (RFC 7719の更新)
- RFC 8552, 2019/3/20 Scoped Interpretation of DNS Resource Records through "Underscored" Naming of Attribute Leaves
- RFC 8553, 2019/3/20: DNS Attrleaf Changes: Fixing Specifications That Use Underscored Node Names
- RFC 8618, 2019/9/3: Compacted-DNS (C-DNS): A Format for DNS Packet Capture
- RFC 8624, 2019/6/11: Algorithm Implementation Requirements and Usage Guidance for DNSSEC
- RFC 8749, 2020/3/27: Moving DNSSEC Lookaside Validation (DLV) to Historic Status
- RFC 8767, 2020/3/31: Serving Stale Data to Improve DNS Resiliency
- RFC 8806, 2020/6/18: Running a Root Server Local to a Resolver (RFC7706の更新)
- RFC 8901, 2020/9/24: Multi-Signer DNSSEC Models
- RFC 8906, 2020/9/22: A Common Operational Problem in DNS Servers: Failure to Communicate
- RFC 8914, 2020/10/23: Extended DNS Errors

RFC 8624 Algorithm Implementation Requirements and Usage Guidance for DNSSEC

- 現在 (2019/6/11) 推奨されるDNSSECアルゴリズム
  - 署名側と検証側それぞれで規定 (アルゴリズム、DS Digest Type)
  - **署名、検証でMUST (実装必須 → おすすめ)**
    - **8(RSASHA256), 13(ECDSAP256SHA256), Digest Type 2 (SHA-256)**
  - **署名、検証でRecommended (実装ほぼ必須): 15(ED25519)**
  - 署名MAY、検証Recommended (普及まだとか、オーバースペック)
    - 14(ECDSAP384SHA384), 16(ED448), Digest Type 4 (SHA-384)
  - 署名Not Recommended、検証MUST (今後使うべきでないもの)
    - 5(RSASHA1), 7(RSASHA1-NSEC3-SHA1), 10(RSASHA512)
  - 署名MUST NOT, 検証MUST (いまある設定は有効): Digest Type 1 (SHA-1)
  - 署名、検証 MUST NOT (使用禁止):
    - 1(RSAMD5), 3(DSA), 6(DSA-NSEC3-SHA1)
  - 署名MUST NOT、検証MAY (ロシアの古い標準規格なのでもう不要)
    - 12(ECC GOST), Digest Type 3(GOST R34.11-94)

RFC 8767: Serving Stale Data to Improve DNS Resiliency

- RFC 8767, 2020/3/31発行
  - DNS耐性向上のための古いデータの提供
  - DoS攻撃などで権威サーバからの応答が得られない場合に、フルサービスリゾルバでキャッシュ有効期限を過ぎたデータを提供する
  - キャッシュから消えて最大7日までのデータをTTL 30で応答する
  - RFC 1034, 1035のTTLの定義、RFC 2181のTTL値の解釈を変更
- すでに実装が進んでいる
  - BIND 9.12以降: stale-answer-enable yes;
  - Unbound 1.8以降: serve-expired: yes
  - Knot resolver: modules = { 'serve_stale < cache' }

ZONEMD RR

- draft-ietf-dnsop-dns-zone-digest で提案、11/19 IESG通過、RFC Editorへ
- ゾーン全体の情報のハッシュ値を保持するZONEMD RRを定義
  - domain TTL IN ZONEMD serial scheme hash_algorithm digest
  - scheme: 1=SIMPLEのみ定義　Hash_algorithm:　1=SHA384, 2=SHA512
- 例: root-servers.net. 3600000 IN ZONEMD 2018091100 1 1 (FEBE…)
- ZONEMDの作成
  - 既存のゾーン頂点のZONEMD RRを削除
  - 空のZONEMD RR追加 (NSEC, NSEC3 type bitmapのため)
  - (DNSSEC署名)
  - DNSSECの順序でゾーン内のリソースレコードを並べかえ
  - 追加したZONEMD以外の、すべてのリソースレコードを連結して、ダイジェストを計算
  - ZONEMD RRを作成 (ZONEMD RRのRRSIGを作成)
- 使用イメージ
  - DNSSEC署名時にZONEごとにZONEMD追加
  - DNSSEC非署名の場合は、プライマリでゾーン情報ロード時に作成
  - ゾーン転送を受け取ったセカンダリサーバでZONEMDをみて確認

SVCB, HTTPS RR

- draft-ietf-dnsop-svcb-https で提案、議論中
  - Service binding and parameter specification via the DNS (DNS SVCB and HTTPS RRs)
- HTTPSなどの接続先情報 (ホスト名、ポート、証明書情報)を保持するSVCB (Service Binding)、HTTPSリソースレコードを定義する
- 目的
  - ドメイン名に対するサービスの複数のエンドポイント情報を保持
  - ドメイン名に複数のSVCB/HTTPSを書いて複数のCDNに対応
    - SVCB/HTTPSごとに異なるCDNサーバ名、サーバ証明書情報
  - ゾーン頂点の別名 (SOA/NSがあるゾーン頂点にはCNAME書けない)
  - Encrypted Server Name Indication(ESNI)で使用する Encrypted ClientHello の key を DNSから得たい
  - 443ではないポート番号の使用
  - HSTSを示す機能 (HTTPS強制)
  - QUICでの接続の強制
  - SVC RRの負荷分散と同じ機能の提供

SVCB, HTTPS RR (2)

- オーナー名 TTL IN SVCB/HTTPS SvcPriority TargetName SvcParams
  - SvcPriorityは16ビットの優先度 (0のときはAliasMode)
  - TargetNameはドメイン名で、CNAMEのような別名を示す　(.の場合はオーナー名)
  - SvcParams はRDATAの残り、(2バイトのキー、2バイトの長さと、バイト列)の列
    - port: ポート番号: 接続先ポート番号指定
    - alpn: TLSのApplication-Layer Protocol Negotiation 指定　　h3=QUIC, h2=HTTP/2
    - echconfig: ESNIで使用するEncrypted ClientHelloの設定データ
    - ipv4hint, ipv6hint: TargetNameの指すIPv4,IPv6アドレス (名前解決を省略)
  - 汎用のSVCBと、HTTPSに特化したHTTPS RR
    - SVCBのオーナー名は _port._protocol.ドメイン名 (SRVと同じ)
    - HTTPS RRのオーナー名はURLのドメイン名部
    - domainname HTTPS は _443._tcp.domainname SVCB に相当するが、HTTPSを使うこと
- 使用例
  - simple.example. 7200 IN HTTPS 1 . alpn=h3　　QUIC対応(HTTP/3)
  - @　7200 IN HTTPS 0 pool.svc.example.　　ゾーン頂点の別名
  - pool　7200 IN HTTPS 1 h3pool alpn=h2,h3 echconfig="123..."　　QUIC/H2+ESNI
    HTTPS 2 .　　alpn=h2 echconfig="abc..."　　HTTP/2のみ+ESNI

SVCB, HTTPS RR (3)

- DNSサーバの変更 (権威サーバ、フルリゾルバ両方)
  - SVCB, HTTPS の TargetName のA, AAAAを知っていたら、Additional sectionに追加する (MXと同じ、メールサーバのA/AAAAを追加する)
- クライアントの動作 (HTTPSの場合)
  1. クライアントは、webサーバのドメイン名の HTTPS, A, AAAA の問い合わせを同時に送る
  2. HTTPS RR応答がAliasModeなら、TargetNameで1.を行う
  3. HTTPS RR応答がServiceModeなら対応するサーバにHTTPS接続
     - ALPN, ECHConfigなどを使用 (場合によってはQUIC)
  4. HTTPS RR応答がなければA, AAAA応答のアドレスにHTTPS接続
- 著者所属Google, Akamai, 謝辞に Apple, Mozilla, Cloudflare の人達
  - Browser開発組織、CDNの人たちが議論に参加しているため、標準化が進むと (DoHのように) すぐにブラウザとCDNでの実装が進む可能性あり

Extended DNS Error (EDE)

- RFC8914, 2020/10/23
- 従来:DNSヘッダのRCODE 4ビット

0=NoError, 2=ServFail, 3=NameError

- EDNS0ヘッダ中の8ビットの Extended RCODE (既存RCODEの上位8ビット)
  - 使いにくかった: 16=BADVERS
- EDNS0にExtended DNS Errorオプションを追加
  - RCODEでエラーを示し、詳細をEDE
- フォーマット

16ビット: Option-Code 15

16ビット: オプション長

16ビット: INFO-CODE(0から24が定義)

可変長: EXTRA-TEXT: UTF8文字列

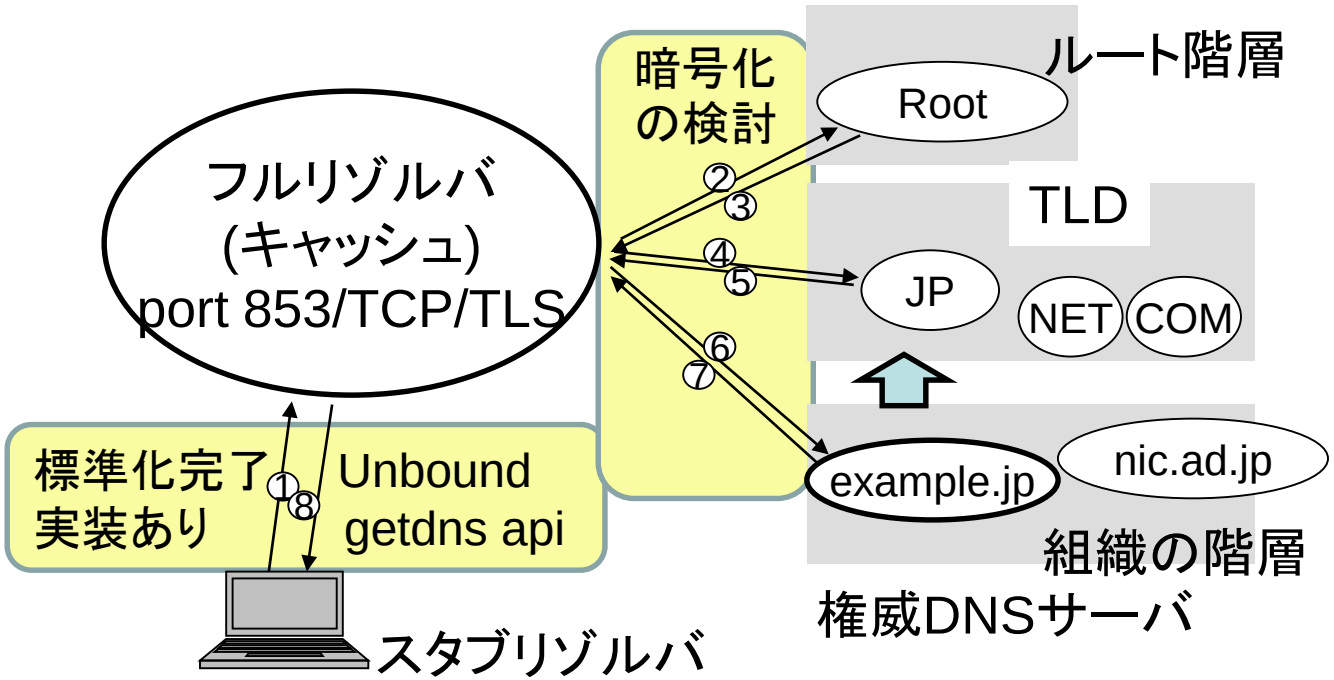
- 近いうちにDNSサーバなどに入る見込み(BIND 9 gitlab mainに若干ある)

0	Other Error		
1	Unsupported DNSSEC Algorithm	13	Cached Error
2	Unsupported DS Digest Type	14	Not Ready
3	Stale Answer	15	Blocked
4	Forged Answer	16	Censored
5	DNSSEC Indeterminate	17	Filtered
6	DNSSEC Bogus	18	Phohibited
7	Signature Expired	19	Stale NXDomain Answer
8	Signature Not Yet Valid	20	Not Authoritative
9	DNSKEY Missing	21	Not Supported
10	RRSIGs Missing	22	No Reachable Authority
11	No Zone Key Bit Set	23	Network Error
12	NSEC Missing	24	Invalid Data

dprive (DNS Privacate Exchange) WG

- スタブリゾルバとフルサービスリゾルバの間の通信を暗号化
- 2014年10月に設立し、ほぼ完了
- RFC 7858 (DNS over TLS)が発行され、使える状態になった
  - 2016/5/17発行
  - 詳細は本日のDoT/DoH入門参照
- DNS over DTLSは使われていない
- DNS over QUICは継続
- ゾーン転送をDNS over TLSで行う拡張の議論は進捗している
  - DNS Zone Transfer-over-TLS
  - サーバ証明書でサーバ名確認など

- IETF 97 (2016/11)にて、フルサービスリゾルバから権威サーバ間の通信暗号化の検討を開始することが提案されたが　IETF 109(2020/11)でも、要求条件の議論を継続



dane (DNS-based Authentication of Named Entities) WG

- DNS(SEC)にTLSの証明書をのせるWG
- 2010年10月設立、標準化を完了し、2017年5月に完了
  - TLSA: RFC 6698, 2012年8月発行、サーバ証明書を載せるRR

    _port._proto.DOMAIN IN TLSA CertUsage Selector MatchingType Certificate_Association_Data
    - CertUsage: 0..CA証明書, 1..エンドエンティティ証明書(CAにより発行), 2..トラストアンカー, 3..自己署名などのエンドエンティティ証明書
    - Selector: 0.. Full certificate, 1.. SubjectPublicKeyInfo
    - MatchingType: 0..証明書そのもの, 1..SHA-256 hash, 2..SHA-512 hash
    - Certificate_association_data: 証明書そのものかhash値、16進表記
  - 例: www.isc.orgのtcpポート443のサーバ証明書問い合わせ

    % dig _443._tcp.www.isc.org TLSA

    443._tcp.www.isc.org. 7185 IN TLSA 3 0 1 7C31F5B6D577A06448C67BAE690E1A3905CA34146BDA86C664EB2690 710D085C

dane WG (2)

- OpenPGPKEY: RFC 7929, 2016年8月発行, Experimental
  - hex(先頭28バイト(sha256(localpart)))._openpgpkey.domain IN OPENPGPKEY 証明書
  - 例: hugh@example.com のOpenPGP証明書をのせる場合
  - c93f1e400f26708f98cb19d936620da35eec8f72e57f9eec01c1afd6._openpgpkey.example.com IN OPENPGPKEY mQCNAzIG[...]
  - PGP Key serverではなく、メールアドレスに対応するDNSクエリでOpenPGP証明書を得る
  - メールアドレスは、大文字小文字どちらでも許容されるため、localpartを一位に決めにくく、議論がまとまらず、Experimental (実験的)プロトコルとして標準化された
- SMIMEA: RFC 8162, 2017年5月発行, Experimental
  - S/MIMEクライアント証明書をのせるもの
  - hex(先頭28バイト(sha256(localpart)))._smimecert.domain IN SMIMEA (TLSAと同じ形式)
- 2017年5月に完了

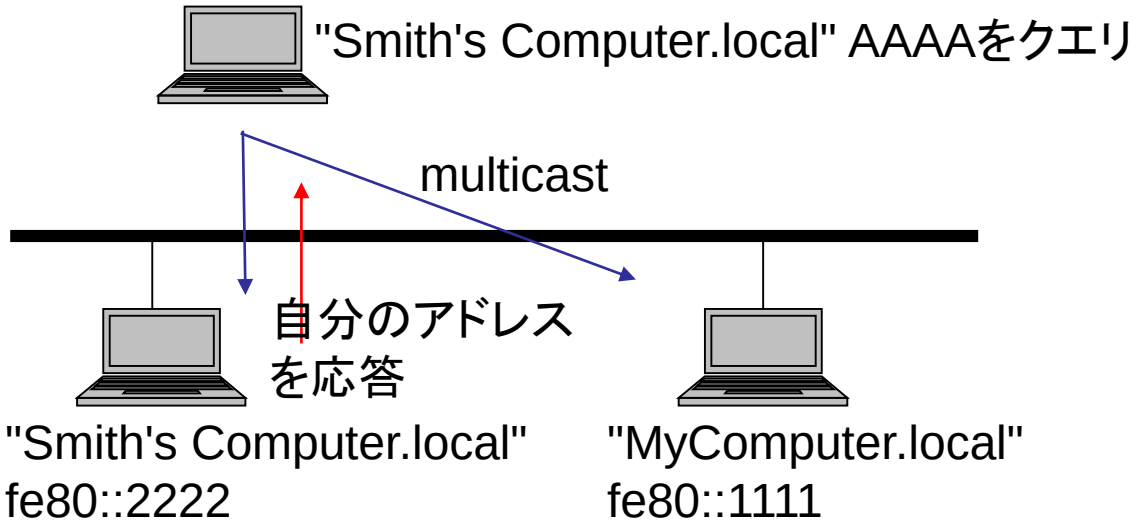
dnssd (Extensions for Scalable DNS Service Discovery) WG

- DNSを使ったサービスディスカバリを作るWG
  - Multicast DNS (RFC 6762, mDNS)とDNS-SD (RFC 6763)をベースに、複数ネットワークセグメントに対応したものを標準化
  - 主にApple社のBonjourとAvahiとして実装されているプロトコルをIETFで標準化したプロトコルにするために拡張を行っている
- Multicast DNS (RFC 6762)
  - link-localでのDNS-likeな名前解決機構
- DNS-SD (RFC 6763)
  - サービスディスカバリ

dnssd: Multicast DNS (RFC 6762)

- link-localでのDNS-likeな名前解決機構
- 各ノードがラベル一つの名前を持ち、.local TLDを用いることでDNSと共存
  - MyComputer.local
  - スペースや' UTF-8も許容
- 各ノードは、multicastでクエリ
  - 224.0.0.251. ff02::fb　port 5353　UDP
  - パケットフォーマットはDNSと同じ
- 各ノードは、自分のホスト名宛クエリを受け取ると、ホスト名とIPアドレスの対応を応答
- 169.254.0.0/16, fe80::/10の逆引き

- Apple社のOSや、Avahiが対応
  - Avahi - Service Discovery for Linux using mDNS/DNS-SD -- compatible with Bonjour



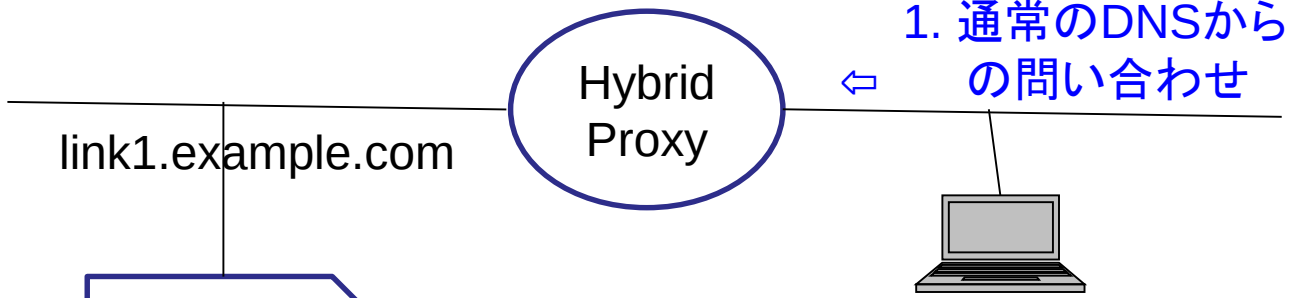
dnssd: DNS-Based Service Discovery (RFC 6763)

- 構造化されたサービス名
  - <Instance>.<Service>.<Domain>
  - SRVと同じ形式 (_sip._udp.domain)
  - ホスト名と違い、スペースやUTF-8許可
- サービスの列挙 (enumeration)
  - サービス名に PTRを書き、サービスを列挙
  - _http._tcp.dns-sd.org PTR ¥032*¥032eBay,¥032online¥032auctions._http._tcp.dns-sd.org.
- サービスへのアクセス
  - SRV RRを使用
  - _http._tcp.dns-sd.org. SRV 0 100 80 www.dns-sd.org.
- Well known service
  - b._dns-sd._udp.domain PTR
    ブラウズ: ドメイン名にあるサービスを列挙
  - db._dns_sd._udp.domain PTR
    ブラウズすべき既定ドメイン名
  - r._dns_sd._udp.domain PTR
    DNSSDの登録サービス (Dynamic Update)
  - dr._dns_sd._udp.domain PTR
    登録サービスの規定ドメイン名
  - lb._dns_sd._udp.domain PTR
    legacy browsingドメイン名: ???

- Multicast DNSでのDNS-SD
  - domain = .local
  - _ipp._tcp.local PTRクエリに対して、同じリンクにある別の名前を持つ複数のプリンタが応答
    - _ipp._tcp.local PTR color._ipp._tcp.local
    - _ipp._tcp.local PTR mono._ipp._tcp.local
  - User Interfaceでcolorを選ぶ、
  - color._ipp._tcp.local 0 0 49152 SRV color.local.
  - color.local IN A 192.0.2.11
  - 192.0.2.11 ポート 49152 に接続

dnssd: コアプロトコル

- RFC 8766, 2020/6/22発行
  - Discovery Proxy for Multicast DNS-Based Service Discovery
  - mDNSとDNSのHybrid proxy
  - リンクごとにドメイン名を設定、ルータなどでproxyを動かす
    - 例: link1.example.com
    - Proxy link1.local ⇔ link1.example.com
    - インターネットから <name>.link1.example.com PTR クエリを受け取ると、<name>.local PTRクエリをlink1でmDNSで送り、応答を書き換えて <name>.link1.example.com応答として返す
  - ブラウズ設定、リンクのドメイン名の設定を管理者が行う
    - b._dns-sd._udp.example.com PTR link1.example.com
    - _link1.example.com でHybrid Proxy



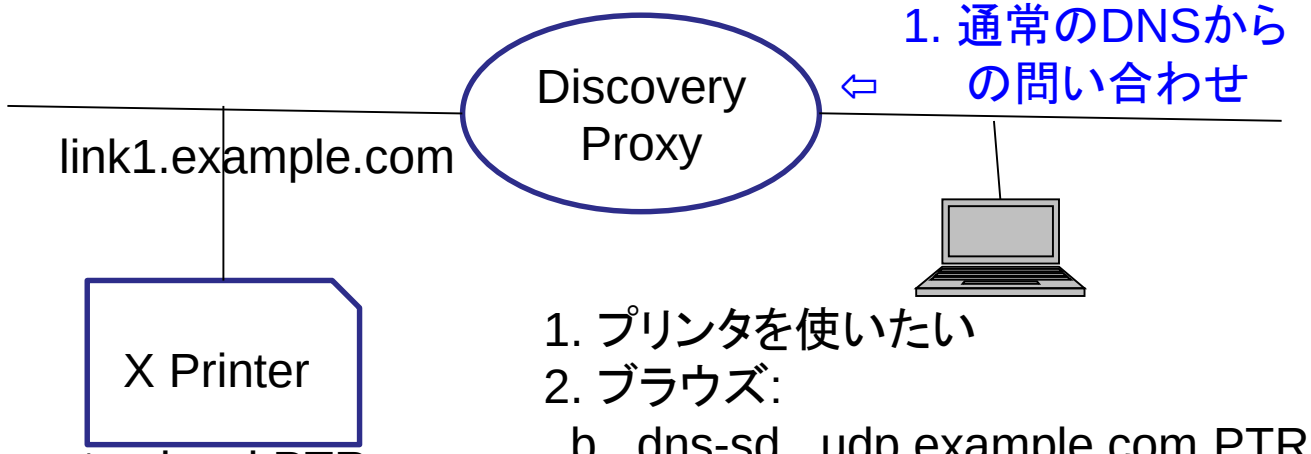
_ipp._tcp.local PTR
"X Printer._ipp._tcp.local"
"X Printer._ipp._tcp.local
SRV ….

1. プリンタを使いたい
2. ブラウズ:
b._dns-sd._udp.example.com
→ link1がある
3. _ipp._tcp.link1.example.com を調べる
4. Proxyが_ipp._tcp.localクエリに変換して問い合わせ
5. プリンタ情報を得られる

dnssd: 追加プロトコル

- RFC 8766 Discovery Proxyはキャッシュを持ち、担当するゾーン情報を管理
  - 例では link1.example.comゾーン
  - Multicast DNSの応答のキャッシュデータ
- RFC 8764, 2020/6/22発行
  - Apple社のDNS Long-Lived-Queries
  - 2005-2020にBonjourで実装・使用したもの
  - RFC 8765で置き換える
- RFC 8765, 2020/6/22発行
  - DNS Push Notifications: Discovery Proxyの管理するゾーンの変更を通知
  - クライアントはDiscovery ProxyのDNSサーバにつなぎっぱなしにしておくと、Discovery Proxyが管理するゾーン情報が変化する度にクライアントに変更点を送る
  - RFC 8490使用
- Apple社はBonjourのプロトコルをIETFのプロトコルとして標準化
  - 著者はAppleのS. Cheshire氏 (mDNS, DNSSD)

- RFC 8490, 2019/3/15発行
  - dnsop WGで標準化
  - DNS Stateful Operations
  - DNSの全く新しい接続方式で、クライアントはサーバとつなぎっぱなしで新しいDSOメッセージをやり取り
  - 新しいOpcode 6 (0 Query, 4 Notify, 5 Update)



_ipp._tcp.local PTR
"X Printer._ipp._tcp.local"
"X Printer._ipp._tcp.local
SRV ….

1. プリンタを使いたい
2. ブラウズ:
b._dns-sd._udp.example.com PTR
→ link1がある
3. _ipp._tcp.link1.example.com PTR を調べる
4. Proxyが_ipp._tcp.local PTR クエリに変換して問い合わせ
5. プリンタ情報を得られる

dnssd: 残務

- 提案されているプロトコル
  - draft-ietf-dnssd-mdns-relay (著者はAppleの人たち)
    - Discovery Proxyの機能を離れたセグメントにリレーするプロトコル (軽量化Discovery Proxy)
  - draft-ietf-dnssd-srp: DNSSDサービス登録プロトコル (著者はAppleの人たち)
    - Discovery ProxyのDNSサーバにDNS Dynamic Updateで情報を登録する
    - Multicast DNSに対応していなくても、Discovery ProxyがMulticast DNSに答えてくれる
- dnssdプライバシーの検討
  - Multicast DNS (RFC 6762) では、マルチキャストが届く範囲(同じイーサネット、同じWiFiなど)では誰からでも問い合わせ、応答が見える
  - 自分のデバイスもdnssdで扱いたいけど、他人からは見られたくないという要求
  - RFC 8882, 2020/9/10発行: DNSSDプライバシーとセキュリティの要求仕様
  - 見られたくないクライアントに暗証番号を指定するなどの提案はあったが難しいので議論が止まっている

最近の提案 (宣伝)

draft-ietf-dnsop-avoid-fragmentation

- 動機
  - IP Fragmentationを使ったキャッシュ汚染攻撃が話題になった
  - EDNS0でIP Fragmentationを使って大きなパケットを返せるようにしたことが間違いだった
    - EDNS0 RFC著者のPaul Vixie氏と共著 (EDNS0ではIP Fragmentationを許容して大きなDNSデータをUDPで扱えるようにした)
- 提案
  - UDP応答ではIP_DONTFRAG, IPV6_DONTFRAGオプションをつけて応答を送る → フラグメントしそうなときはエラー
  - 問い合わせを送る側、応答する側双方で、Path MTUを意識して、フラグメントしない範囲でDNS応答をUDPで送る
  - 問い合わせを送る側は EDNS0 での UDPサイズをPath MTUからIPヘッダ、UDPヘッダを引いたサイズ以内とする
    - DNS Flag Day のひとたちは、1232 を推奨している
- 現状: dnsop WGで、かなり好意的に扱われている

draft-fujiwara-dnsop-delegation-information-signer

- 概要: DNSSECで、委任情報に署名を追加する提案
- 動機
  - DNSSECでは、委任情報は署名されない (攻撃されても防御できない)
  - TLDなどでは、顧客から委任情報を受け取る (ネームサーバ情報とホスト情報とDS)が署名対象はDSだけ
- 提案
  - 委任情報の親側のNSと、内部名(in-domain)グルーA/AAAAは一意に決まるので、DSを転用してそこにハッシュ値を入れる
  - SHA-256 hash(parent side NS RRSet | in-domain glue records)
  - NS RRSet, in-domain glue recordsはDNSSEC canonical orderで並べる
  - 例: example.jp IN DS 0 0 XX _SHA256_hash(NS|glue)
  - DS RRSetとして署名
  - DNSSEC ValidatorがDigest type XXを受け取ったら委任情報を検証し、一致しなければ捨てる

An example of DiS record generation

- dig +norec +dnssec @a.dns.jp wide.ad.jp

;; AUTHORITY SECTION:
wide.ad.jp. 86400 IN NS ns.tokyo.wide.ad.jp.
wide.ad.jp. 86400 IN NS ns-wide.wide.ad.jp.
wide.ad.jp. 86400 IN NS mango.itojun.org.
wide.ad.jp. 7200 IN DS 32584 8 2 1D7EEF8BC...
wide.ad.jp. 7200 IN RRSIG DS …
;; ADDITIONAL SECTION:
ns.tokyo.wide.ad.jp. 86400 IN AAAA 2001:200:0:1::6
ns-wide.wide.ad.jp. 86400 IN AAAA 2001:200:0:1::f
ns.tokyo.wide.ad.jp. 86400 IN A 203.178.136.35
ns-wide.wide.ad.jp. 86400 IN A 203.178.136.59

1. Remove old DiS
2. Generate new DiS
2.1 Collect referral NS RRSet and in-domain glue
2.2 Reorder NS RRSet and in-domain glue as DNSSEC canonical order [RFC 4034]
2.3 Calculate SHA-256 hash
SHA-256(
wide.ad.jp. 86400 IN NS ns.tokyo.wide.ad.jp.
wide.ad.jp. 86400 IN NS ns-wide.wide.ad.jp.
wide.ad.jp. 86400 IN NS mango.itojun.org.
ns-wide.wide.ad.jp. 86400 IN A 203.178.136.59
ns-wide.wide.ad.jp. 86400 IN AAAA 2001:200:0:1::f
ns.tokyo.wide.ad.jp. 86400 IN A 203.178.136.35
ns.tokyo.wide.ad.jp. 86400 IN AAAA 2001:200:0:1::6)
2.4 Generated DiS data
wide.ad.jp 7200 IN DS 0 0 XX _SHA256_hash(NS|glue)
3. Sign DS RRSet (contains generated DiS and original DS)

draft-fujiwara-dnsop-delegation-information-signer (4)

- IETF 109でいただいたコメント
  - なぜ、委任情報を守らなかったか？
    → DNSへの変更を最小限としたため (元dnsext WG co-chair の Olaf Kolkman氏より)
  - DNS関係者からは、嬉しさがわかりにくい
    - DNSSEC検証すれば、案内された委任先でエラーにできるため
  - dprive WGの人たちからは、興味を持たれた
    - DSを利用するアイデアは、IETF 108でのdprive WGでの提案 draft-vandijk-dprive-ds-dot-signal-and-pin を参考にしたもの
  - 無視されたり討ち死にするつもりだったのに、時間をもらえ、コメントももらえてよかった

まとめ

- dnsop WG
  - 名前解決の効率化や攻撃耐性の強化、新機能追加のための拡張が盛んに行なわれ、実装も進む
  - DNSソフトウェア開発者、ブラウザ開発者、CDNなどの開発者が多数集まっている
- dprive WG
  - クライアントからフルリゾルバ間の通信路暗号化の標準化は完了し、すでに使用可能
  - 今後、フルリゾルバから権威DNSサーバ間の暗号化に取り組む
  - ゾーン転送の暗号化に取り組む
- dane WG
  - サーバ証明書と、OpenPGP、SMIMEの個人証明書をDNSに載せることができるようになった
  - 今後ブラウザやメールソフトウェアでの実装が期待される
- dnssd
  - Multicast DNSを複数セグメントで使用する拡張が標準化された
  - Apple社のOSで実装されている
- IETF
  - 既存のプロトコルの問題や、新しい提案は歓迎される

参考

- www.ietf.org → datatracker.ietf.org
  - IETFミーティングの資料、議事録
  - ワーキングループの情報
    - 標準化したRFCへのリンク
    - 議論中のdraftへのリンクや状態
    - メーリングリストアーカイブ
- www.rfc-editor.org
  - RFC